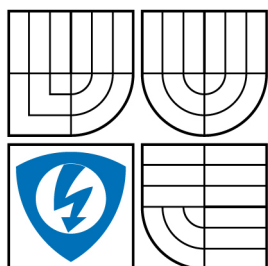


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
**BRNO UNIVERSITY OF TECHNOLOGY**



**FAKULTA ELEKTROTECHNIKY A**  
**KOMUNIKAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV RADIOELEKTRONIKY**

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

# **SROVNÁNÍ KOMPRIMAČNÍCH ALGORITMŮ PRO VIDEO VE VYSOKÉM ROZLIŠENÍ**

COMPARISON OF COMPRESSION ALGORITHMS FOR HIGH DEFINITION VIDEO

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

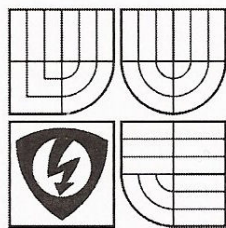
**AUTOR PRÁCE**  
AUTHOR

Martin Dvořák

**VEDOUCÍ PRÁCE**  
SUPERVISOR

Ing. Martin Slanina, Ph.D.

BRNO 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Bakalářská práce

bakalářský studijní obor  
Elektronika a sdělovací technika

**Student:** Martin Dvořák

**Ročník:** 3

**ID:** 106418

**Akademický rok:** 2009/10

## NÁZEV TÉMATU:

**Srovnání komprimačních algoritmů pro video ve vysokém rozlišení**

## POKYNY PRO VYPRACOVÁNÍ:

Prostudujte komprimační algoritmy, které se používají pro digitální video ve vysokém rozlišení. Analyzujte dílčí nástroje jednotlivých kodeků. Vytvořte databázi videosekvencí ve vysokém rozlišení komprimovaných s využitím různých kodérů s různým nastavením.

Porovnejte jednotlivé komprimační algoritmy co do výkonnosti (závislost kompresního poměru na objektivně měřené kvalitě) a subjektivní kvality výsledného videa (typická zkrácení generovaná jednotlivými algoritmy).

## DOPORUČENÁ LITERATURA:

[1] RICHARDSON, I. E. G. H.264 And MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia. Chichester: J. Wiley & Sons, 2003.

**Termín zadání:** 8.2.2010

**Termín odevzdání:** 28.5.2010

**Vedoucí práce:** Ing. Martin Slanina, Ph.D.

**Konzultanti bakalářské práce:**

**prof. Dr. Ing. Zbyněk Raida**  
předseda oborové rady



## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

# LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

## 1. Pan/paní

Jméno a příjmení: Martin Dvořák  
Bytem: Drahanovice 176, 783 44  
Narozen/a (datum a místo): 10. dubna 1987 v Olomouci

(dále jen „autor“)

a

## 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií  
se sídlem Údolní 53, Brno, 602 00  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:  
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika  
(dále jen „nabyvatel“)

## Čl. 1

### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
- ☐ diplomová práce
- ☒ bakalářská práce
- ☐ jiná práce, jejíž druh je specifikován jako .....  
(dále jen VŠKP nebo dílo)

Název VŠKP: Srovnání komprimačních algoritmů pro video ve vysokém rozlišení

Vedoucí/ školitel VŠKP: Ing. Martin Slanina, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: \_\_\_\_\_

VŠKP odevzdal autor nabyvateli\*:

- ☒ v tištěné formě – počet exemplářů: 2
- ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

---

\* hodící se zaškrtněte

## Článek 2

### Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☒ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3

### Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 28. května 2010

.....  
Nabyvatel

.....  
Autor

# **Abstrakt**

Bakalářské práce se zabývá srovnáním a analýzou jednotlivých nástrojů u současných komprimačních algoritmů využívaných pro komprimaci videa ve vysokém rozlišení. Sekvence videosignálů vyžadují velké množství paměti pro jejich uchovávání a zabírají velkou šířku pásma při jejich přenosu, proto je nutné videosignály komprimovat a to s ohledem na kvalitu videosignálu po dekomprimaci. Práce se zaměřuje hlavně na rozdíly mezi jednotlivými algoritmy a celkově jejich funkcí. Součástí práce je databáze komprimovaných videosekvencí, porovnání výkonnosti jednotlivých algoritmů a srovnání subjektivní kvality komprimovaných videosekvencí.

## **Klíčová slova**

Komprimační algoritmus, MPEG-2, MPEG-4 Part 10, H.264, AVC, VC-1, WMV9, Předpověď, Diskrétní kosinova transformace, DCT, kvantování, komprese videa, rekonstrukční filtr, vektor pohybu, skupina snímků, GOP, PSNR, index SSIM.

# **Abstract**

Bachelor's thesis deals with comparison and analysis of various instruments of the present-day compression algorithms used to compress high-definition video. Video sequences require a large amount of memory for their storage and take up bandwidth for their transmission, so it is necessary to compress video signals with regard to the quality of video after decompressing. The work focuses mainly on the differences between the algorithms and overall performance. Part of this work is a compressed video database, comparing the performance of different algorithms and the subjective quality of compressed video.

## **Keywords**

Compression algorithm, MPEG-2, MPEG-4 Part 10, H.264, AVC, VC-1, WMV9, Prediction, Discrete cosine transform, DCT, quantisation, video compression, deblocking filter, motion vector, group of pictures, GOP, PSNR, index SSIM, PSNR, index SSIM.

DVOŘÁK, M. *Srovnání komprimačních algoritmů pro video ve vysokém rozlišení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 67 s. Vedoucí semestrální práce Ing. Martin Slanina, Ph.D.

# Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma Srovnání komprimačních algoritmů pro video ve vysokém rozlišení jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 28. května 2010

.....  
podpis autora

# Poděkování

Děkuji vedoucímu bakalářské práce Ing. Martinu Slaninovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V Brně dne 28. května 2010

.....  
podpis autora

# OBSAH

|   |    |
|---|----|
| Seznam obrázků .....  | 9  |
| Seznam tabulek .....  | 11 |
| 1 Úvod .....  | 12 |
| 2 MPEG-2 .....  | 13 |
| 2.1 Blokové zapojení kodéru .....                               | 13 |
| 2.2 Kódování uvnitř snímku (Intra-frame) .....                  | 14 |
| 2.2.1 Transformační kódování (Transform) .....                  | 14 |
| 2.2.2 Kvantování (Quantization) .....                           | 16 |
| 2.3 Mezisnímkové kódování (Inter-frame) .....                   | 17 |
| 2.3.1 Druhy snímků a jejich předpověď .....                     | 17 |
| 2.3.2 Skupina snímků (Group of Pictures) .....                  | 18 |
| 2.3.3 Vektor pohybu (Motion vector) .....                       | 18 |
| 2.4 Entropické kódování .....                                   | 19 |
| 2.5 Přehled profilů a úrovní standardu MPEG-2 .....             | 20 |
| 3 MPEG-4 Part 10 (H.264/AVC) .....                              | 21 |
| 3.1 Blokové zapojení kodéru .....                               | 21 |
| 3.2 Předpověď (Prediction) .....                                | 22 |
| 3.2.1 Přepověď uvnitř snímku (Intra prediction) .....           | 23 |
| 3.2.2 Přepověď mezi snímky (Inter prediction) .....             | 25 |
| 3.3 Rekonstrukční filtr (Deblocking Filter) .....               | 27 |
| 3.4 Transformační kódování (Transform) .....                    | 28 |
| 3.5 Přehled profilů .....                                       | 29 |
| 4 VC-1 (WMV9) .....   | 30 |
| 4.1 Blokové zapojení kodéru .....                               | 30 |
| 4.2 Předpověď uvnitř snímku (Intra prediction) .....            | 31 |
| 4.3 Předpověď mezi snímky (Inter prediction) .....              | 31 |
| 4.3.2 Interpolace (Interpolation) .....                         | 32 |
| 4.3.3 Předpověď vektoru pohybu (Motion Vector Predictors) ..... | 32 |
| 4.4 Skupina snímků (Group of Pictures) .....                    | 33 |
| 4.5 Transformační kódování (Transform) .....                    | 34 |
| 4.6 Rekonstrukční filtr (Deblocking Filter) .....               | 34 |
| 4.7 Přehled profilů .....                                       | 35 |
| 5 Metriky pro měření kvality .....                              | 36 |
| 6 Databáze komprimovaných videosekvencí .....                   | 38 |
| 7 Porovnání výkonnosti algoritmů .....                          | 44 |
| 7.1 Testovací videosekvence formátu 720p .....                  | 44 |
| 7.1.1 Závislosti PSNR jasové složky Y na bitové rychlosti ..... | 44 |
| 7.1.2 Závislosti indexu SSIM na bitové rychlosti .....          | 48 |
| 7.1.3 Porovnání subjektivní kvality .....                       | 52 |
| 7.2 Testovací videosekvence formátu 1080p .....                 | 55 |
| 7.2.1 Závislosti PSNR jasové složky Y na bitové rychlosti ..... | 55 |
| 7.2.2 Závislosti indexu SSIM na bitové rychlosti .....          | 59 |
| 7.1.3 Porovnání subjektivní kvality .....                       | 62 |
| 8 Závěr .....   | 64 |
| Seznam literatury .....   | 65 |
| Seznam příloh .....   | 67 |



# Seznam obrázků

|  |    |
|--|----|
| Obr. 1. Blokové zapojení kodéru MPEG-2 [1].  | 13 |
| Obr. 2. DCT bloku vzorků s malými rozdíly [1].   | 15 |
| Obr. 3. DCT bloku vzorků s velkými rozdíly [1].  | 15 |
| Obr. 4. Nelineární kvantování frekvenčních koeficientů [1].                                      | 17 |
| Obr. 5. Skupina snímků GOP [1].  | 18 |
| Obr. 6. Čtení „cik-cak“ kvantovaných koeficientů v matici 8x8 [3].                               | 19 |
| Obr. 7. Blokové zapojení kodéru MPEG-4 AVC [9].  | 21 |
| Obr. 8. Předpověď uvnitř snímku [9].   | 22 |
| Obr. 9. Předpověď mezi snímky [9].   | 22 |
| Obr. 10. Předpověď jasového bloku 4x4 [9].   | 23 |
| Obr. 11. Označení vzorků předpovědi [9].   | 23 |
| Obr. 12. 4x4 jasové módy předpovědi [10].  | 24 |
| Obr. 13. 16x16 jasové módy předpovědi [10].  | 24 |
| Obr. 14. Stromová struktura [9].   | 25 |
| Obr. 15. Interpolace hodnot vzorků v poloze $\frac{1}{2}$ pixelu [9].                            | 26 |
| Obr. 16. Interpolace hodnot vzorků v poloze $\frac{1}{4}$ pixelu [9].                            | 26 |
| Obr. 17. Předpověď bloku E ze: a) stejné velikosti bloků b) rozdílné velikosti bloků [9].        | 27 |
| Obr. 18. Pořadí filtrování v makrobloku a) jasové složky b) barevné složky [9].                  | 28 |
| Obr. 19. Vzorky v blízkosti svislé a vodorovné hranice [9].                                      | 28 |
| Obr. 20. Transformační kódování MPEG-4 AVC [9].  | 29 |
| Obr. 21. Blokové zapojení kodéru VC-1 [20].  | 30 |
| Obr. 22. Intra DC/AC předpověď [19].   | 31 |
| Obr. 23. Bloky pohybové kompenzace VC-1 [19].  | 32 |
| Obr. 24. Předpověď vektoru pohybu aktuálního makrobloku [19].                                    | 33 |
| Obr. 25. Skupina snímků s volnou strukturou [19].  | 33 |
| Obr. 26. Definice hranic pro filtrování [19].  | 34 |
| Obr. 27. Systém měření strukturální podobnosti [15].   | 37 |
| Obr. 28. Ukázka snímku ze sekvence ParkJoy s obtížnou náročností kódování.                       | 38 |
| Obr. 29. Ukázka snímku ze sekvence OldTownCross s jednoduchou náročností kódování.               | 39 |
| Obr. 30. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci CrowdRun.     | 46 |
| Obr. 31. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci ParkJoy.      | 46 |
| Obr. 32. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci DucksTakeOff. | 47 |
| Obr. 33. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci InToTree.     | 47 |
| Obr. 34. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci OldTownCross. | 48 |
| Obr. 35. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci CrowdRun.              | 50 |
| Obr. 36. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci ParkJoy.               | 50 |
| Obr. 37. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci DucksTakeOff.          | 51 |
| Obr. 38. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci InToTree.              | 51 |
| Obr. 39. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci OldTownCross.          | 52 |
| Obr. 40. Originální snímek s vybranými oblastmi pro porovnání – 720p.                            | 52 |
| Obr. 41. Porovnání kvality snímků – oblast 1 a) originální b) AVC c) VC-1 d) MPEG-2.             | 53 |
| Obr. 42. Porovnání kvality snímků – oblast 2 a) originální b) AVC c) VC-1 d) MPEG-2.             | 53 |
| Obr. 43. Porovnání kvality snímků – oblast 1 a) originální b) AVC c) VC-1 d) MPEG-2.             | 54 |

|   |    |
|---|----|
| Obr. 44. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci CrowdRun. ....     | 56 |
| Obr. 45. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci ParkJoy. ....      | 57 |
| Obr. 46. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci DucksTakeOff. .... | 57 |
| Obr. 47. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci InToTree. ....     | 58 |
| Obr. 48. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci OldTownCross. .... | 58 |
| Obr. 49. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci CrowdRun. ....              | 60 |
| Obr. 50. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci ParkJoy. ....               | 60 |
| Obr. 51. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci DucksTakeOff. ....          | 61 |
| Obr. 52. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci InToTree. ....              | 61 |
| Obr. 53. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci OldTownCross. ....          | 62 |
| Obr. 54. Originální snímek s vybranými oblastmi pro porovnání – 1080p. ....                           | 62 |
| Obr. 55. Porovnání kvality snímků a) originální b) AVC c) VC-1 d) MPEG-2. ....                        | 63 |

## Seznam tabulek

|  |    |
|--|----|
| Tab. 1. Profily a úrovně standardu MPEG-2 [1]. .....                               | 20 |
| Tab. 2. Přehled profilů a dostupných nástrojů AVC [9]. .....                       | 29 |
| Tab. 3. Přehled profilů a dostupných nástrojů VC-1 [19]. .....                     | 35 |
| Tab. 4. Nastavení rekonstrukčního filtru AVC. ....                                 | 39 |
| Tab. 5. Výpis komprimovaných sekvencí pro formát 720p. ....                        | 40 |
| Tab. 6. Výpis komprimovaných sekvencí pro formát 1080p. ....                       | 41 |
| Tab. 7. Nastavení profilů algoritmu VC-1. ....                                     | 42 |
| Tab. 8. Nastavení profilů algoritmu AVC. ....                                      | 43 |
| Tab. 9. Srovnání bitové rychlosti algoritmů při shodné kvalitě (PSNR/720p). ....   | 45 |
| Tab. 10. Srovnání bitové rychlosti algoritmů při shodné kvalitě (SSIM/720p). ....  | 49 |
| Tab. 11. Srovnání bitové rychlosti algoritmů při shodné kvalitě (PSNR/1080p). .... | 56 |
| Tab. 12. Srovnání bitové rychlosti algoritmů při shodné kvalitě (SSIM/1080p). .... | 59 |

# 1 Úvod

Komprimační algoritmy slouží ke snížení datového toku a tím i velikosti výsledného souboru u digitálně zpracovávaných videozáznamů při co nejmenším viditelném zhoršení kvality po dekódování. Mezi nejpoužívanější komprimační algoritmy pro videozáznamy patří v současné době MPEG-2, MPEG-4, VC-1. Standardy MPEG-2, MPEG-4 obsahují mnoho částí, tzv. partů, které se zabývají rozdílnými úkoly. Tato práce se zabývá pouze komprimací videosekvencí a budou zde popisovány části zabývající se právě kódováním videosekvencí – MPEG-2 Part 2 a Part 10 u MPEG-4.

Písmena MPEG jsou zkratkou **M**otion **P**ictures **E**xperts **G**roup, což je skupina spolupracujících s ISO - **I**nternational **O**rganisation for **S**tandardisation, a zabývá se vývojem mezinárodních standardů pro kompresi, dekompresi, zpracování a kódování pohyblivých obrázků, zvuků a jejich kombinací s cílem pokrýt požadavky širokého spektra aplikací.

MPEG-2 (ISO/IEC 13818) byl představen v roce 1994, respektive byly schváleny první 3 části – **S**ystém, **V**ideo, **A**udio, a v současné době obsahuje 11 částí. Druhá část MPEG-2 Part 2 (ISO/IEC 13818-2) popisovaná v práci má také označení ITU-T Recommendation H.262 (ITU - **I**nternational **T**elecommunication **U**nion) a popisuje kompresní kodek pro prokládaný a neprokládaný video signál.

MPEG-4 (ISO/IEC 14496) byl představen v roce 1999, respektive byla schválena jeho druhá verze. Standard čítá v současné době 27 částí + 2 částí je vývoji. V práci je popisovaná MPEG-4 Part 10, jiné označení je H.264/AVC. Jedná se o kodek vyvinutý skupinou JVT (**J**oint **V**ideo **T**eam), která vznikla spoluprací ITU-T Video Coding Experts Group (VCEG) spolu s ISO/IEC MPEG. ITU-T H.264 standard a ISO/IEC MPEG-4 AVC standard (formálně ISO/IEC 14496-10 - MPEG-4 Part 10, Advanced Video Coding) mají stejný technický obsah a dokumentaci. H.264 je populární pro jeho použití na Blu-ray discích, HD DVD.

VC-1 je video kodek standardizovaný společností SMPTE (**S**ociety of **M**otion **P**icture and **T**elevision **E**ngineers), VC-1 je označením pro SMPTE 421M. Původně byl vyvinut Microsoftem jako proprietární video formát, než byl vydán jako oficiální SMPTE standard 3. dubna 2006. Dnes je široce podporován na HD DVD, Blu-ray discích, webovou aplikací Silverlight společnosti Microsoft a jako kodek Windows Media Video 9 (WMV9).

Náplní práce je analyzovat používané nástroje jednotlivých komprimačních algoritmů, porovnat je co do výkonnosti a subjektivní kvality. Kapitola č. 2 se věnuje algoritmu MPEG-2. Kapitoly č. 3 a 4 obsahují popis nástrojů a funkcí algoritmu MPEG-4 H.264/AVC a VC-1, které vylepšují kódování oproti algoritmu MPEG-2. Kapitola č. 5 obsahuje popis metrik pro měření kvality. Kapitola č. 6 popisuje databázi komprimovaných sekvencí, přehled nastavení parametrů a profilů. Poslední kapitola se věnuje porovnání výkonnosti algoritmů, jedná se o závislosti PSNR a indexu SSIM na bitové rychlosti. Součástí je také porovnání subjektivní kvality stejného snímku, stejné kvality, komprimovaného jednotlivými algoritmy.

## 2 MPEG-2

MPEG-2 je ztrátový komprimační datový formát. Je nástupcem formátu MPEG-1. MPEG-2 se užívá jako standardní formát pro ukládání a přenos videa na DVD, nebo při distribuci digitálního televizního signálu DVB a v několika případech i u disků Blu-ray. U aplikací, které vyžadují MPEG-2 komprimaci či dekomprimaci videa v reálném čase, jsou kladeny výrazně vyšší nároky na výpočetní kapacitu procesoru, než u formátu MPEG-1 [2].

Je flexibilní a otevřený, norma ISO/IEC 13818 připouští různé snímkové kmitočty i různý počet prokládaných či neprokládaných řádků. Je stabilní, lze volit různé kvalitativní úrovně výsledného obrazu. Další výhodou je nesymetrickost, kdy komplikovaný je pouze kodér, zatímco dekodér je jednoduchý [7].

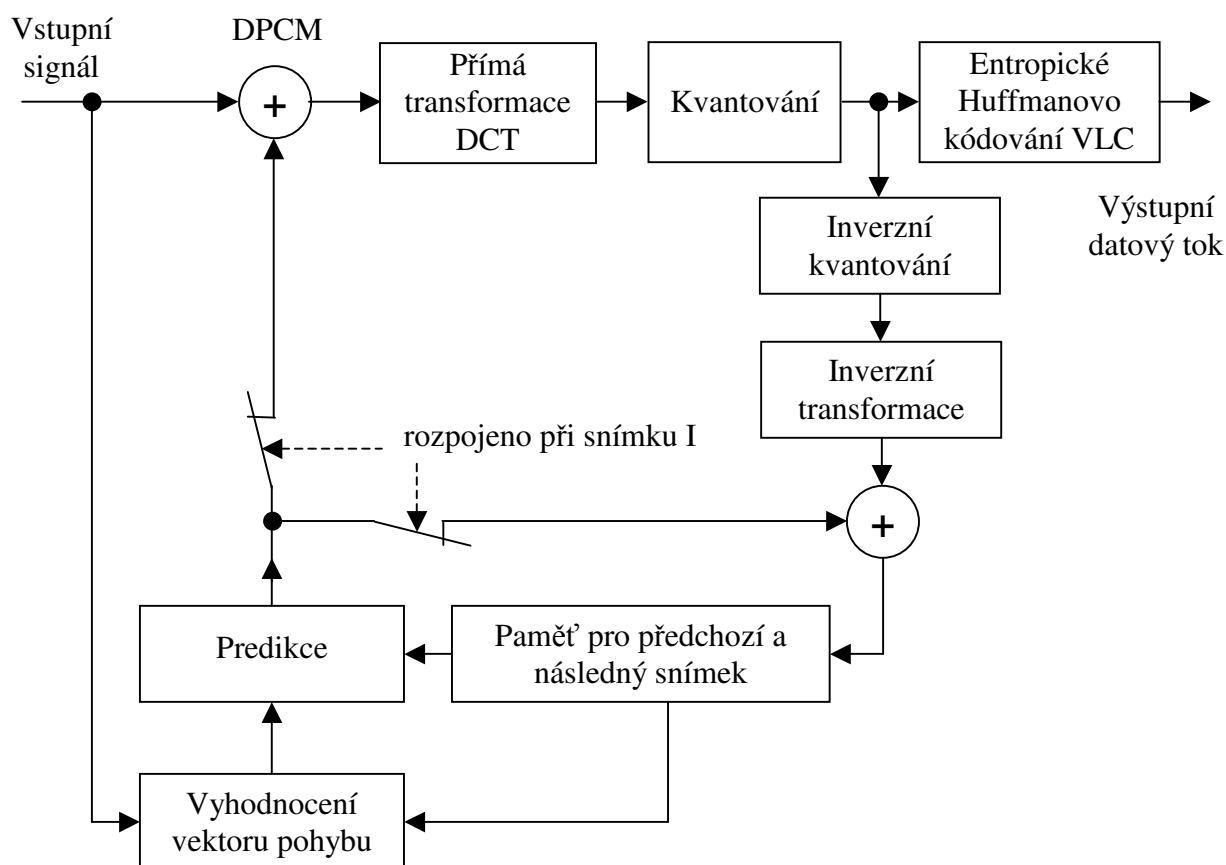
### 2.1 Blokové zapojení kodéru

Kódování u standardu MPEG-2 se dělí na dvě úrovně, kódování uvnitř snímku (Intra-frame) a kódování mezisnímkové (Inter-frame).

Kódování uvnitř snímku je tvořeno bloky, viz obr. 1:

**Přímá transformace DCT** – Převod obrazových vzorků na frekvenční koeficienty.

**Kvantování** – Dělení matice frekvenčních koeficientů kvantizační maticí.



Obr. 1. Blokové zapojení kodéru MPEG-2 [1].

Mezisnímkové kódování je tvořeno bloky:

**Predikce** (Pohybová kompenzace) - Tvorba predikce z vektoru pohybu a předešlého/následného snímku.

**Vyhodnocení vektoru pohybu** (Pohybový odhad) - Porovnávání aktuálního snímku s předchozím (následným), hledání stejných makrobloků, na základě jejich vzájemné pozice se vytváří vektor pohybu.

**Paměť pro předchozí a následný snímek** – Do paměti se ukládá rekonstruovaný snímek a slouží jako reference pro předpověď.

Další bloky:

**DPCM** – Vytváří rozdíl mezi aktuálním a předchozím (následným) snímkem. Při snímku typu I rozdíl nevytváří.

**Inverzí kvantování, inverzní transformace** – Rekonstrukce rozdílového snímku, zpětné násobení kvantované matice a zpětná transformace  $DCT^{-1}$ .

**Entropické Huffmanovo kódování VLC** – „Cik cak“ čtení frekvenčních koeficientů a následné kódování.

## 2.2 Kódování uvnitř snímku (Intra-frame)

Základním principem komprese obrazu je redukce redundantní a irelevantní části informace v obrazovém signálu. Redukce redundance (nadbytečnosti) je proces bezetrátový, redukce resp. odstranění irelevance (zbytečnosti) je proces ztrátový. Při redukci irelevance se využívá znalosti fyziologických vlastností lidského zraku, respektive jeho nedokonalosti, který je málokdy schopen rozeznat drobné detaily v obraze, zvláště při rychlém pohybu, a proto je možno je vynechat [1].

### 2.2.1 Transformační kódování (Transform)

Transformační kódování se využívá pro zmenšení entropie signálu a tím i snížení potřebné bitové rychlosti. Princip spočívá v transformační operaci, kdy dojde k nahrazení prostorového rozložení hodnot vzorků signálu spektrem jeho frekvenčních složek s příslušnými amplitudami zvanými transformační koeficienty, přičemž nulové a jim blízké hodnoty transformačních frekvenčních koeficientů nepřenášíme a tím dojde ke zmenšení počtu přenášených dat, respektive výsledné přenosové rychlosti, s využitím redukce redundance v signálu. Příkladem je diskrétní kosinová transformace, která je základem většiny komprimačních metod [1].

Koeficienty DCT, tj. funkce  $G(u,v)$  se počítají pro blok  $N = 8$  (matice  $8 \times 8$  prvků) z funkce  $g(x,y)$  podle rovnice [1]:

$$G(u,v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^7 \sum_{y=0}^7 g(x,y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}, \quad (1)$$

kde

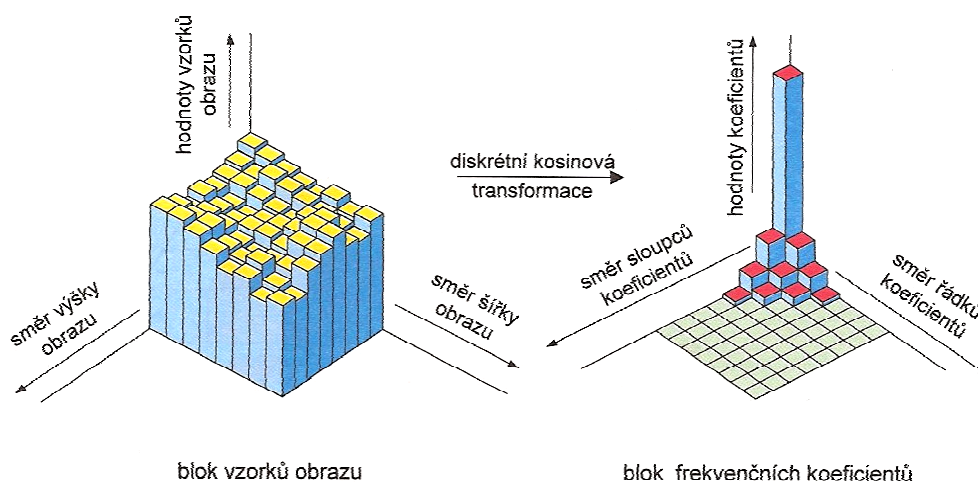
$u, v$  jsou souřadnice ve frekvenční oblasti,

$x, y$  jsou souřadnice v prostorové (časové) oblasti,

konstanty  $C(u)=C(v)=1/\sqrt{2}$  pro  $u = v = 0$ .  
 $C(u)=C(v)=1$  pro  $u > 0, v > 0$ .

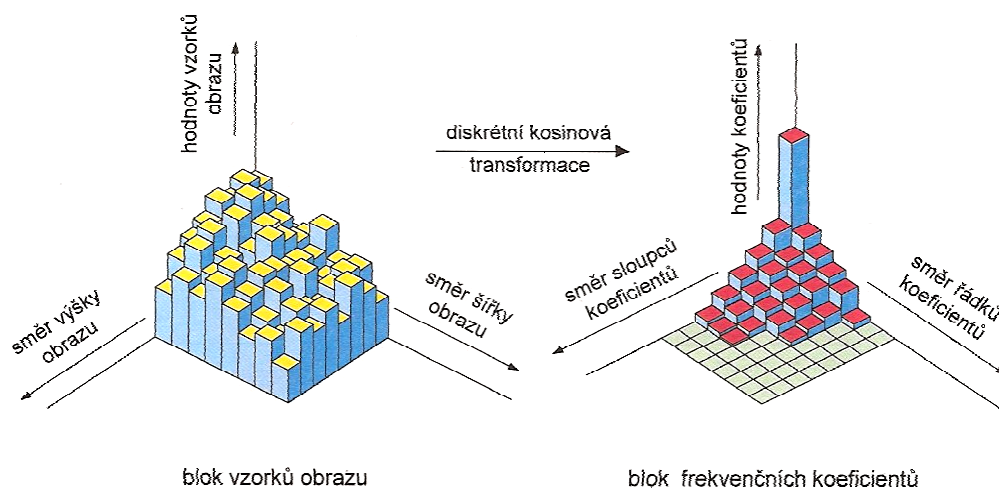
Jmenovatel ve zlomku u funkce  $\cos$  je v obecném případě  $2N$ .

Koeficient  $G(0, 0)$  představuje stejnosměrnou hodnotu, tedy střední hodnotu celého transformovaného signálu. V této hodnotě je soustředěna celá energie signálu, neboť její velikost převyšuje amplitudy ostatních střídavých složek. Hodnoty střídavých složek klesají, s postupně rostoucí frekvencí, od levého rohu doprava a směrem dolů. Čím méně vysokofrekvenčních složek se v obrazu vyskytuje, tím více nul nebo hodnot blízkých nule je ve výsledku. Každý z koeficientů  $G(u,v)$  se počítá ze všech 64 vzorků jednoho bloku [1], [7].



Obr. 2. DCT bloku vzorků s malými rozdíly [1].

Při transformaci matice signálových vzorků na matici frekvenčních koeficientů se jedná o dvourozměrnou transformaci 2D DCT. Matice vzorků se, z důvodu technické náročnosti zpracování, rozděluje na bloky  $8 \times 8$  prvků (u MPEG-2). Nedosáhne se tak maximální vytiženosti transformace, kdy celkový počet vynechaných frekvenčních koeficientů, které jsou nulové nebo blízké nulové hodnotě, není tak veliký jako u transformace celého snímku. Tím dochází i ke zmenšení stupně komprimace [1], [7].



Obr. 3. DCT bloku vzorků s velkými rozdíly [1].

U jednolitých ploch, kde se nevyskytují vyšší frekvence, je tento způsob vhodný (nejvyšší koeficienty budou nulové). Obecně platí, čím menší rozdíly v hodnotách vzorků, tím méně početná skupina frekvenčních koeficientů, viz obr. 2. V nejlepším případě, kdy jsou všechny vstupní hodnoty stejné, dostaneme po DCT pouze jeden koeficient se stejnou hodnotou. Naopak, obraz s prudkými změnami v hodnotách vzorků vytváří početnou skupinu frekvenčních koeficientů, viz obr. 3 [1], [7].

Je zřejmé, že DCT není bezztrátová, neboť vznikají chyby způsobené zaokrouhlováním frekvenčních koeficientů, po inverzní transformaci  $DCT^{-1}$  tedy nedostaneme původní signál PCM. To vše za normálních podmínek. V situaci, kdy použijeme prahování, dojde k zanedbání všech platných hodnot, které jsou dány hranicí prahování. Tím dojde ke zvýšení komprese, ale za cenu více chyb, respektive degradace v obraze [1], [7].

## 2.2.2 Kvantování (Quantization)

Úkolem kvantování je snížení bitové rychlosti. Ke snížení dochází vyřazením vzniklých nulových a nule blízkých koeficientů z přenosu. Kvantování je oproti diskrétní kosinové transformaci úprava ztrátová a využívá nedokonalosti lidského zraku. Při dodržení povolené velikosti ztrát nevyhodnotí lidský zrak ztrátu jako škodlivou [1].

Pro výpočet frekvenčních koeficientů je třeba  $N+3$  bitů, kde  $N=8$  (matice  $8 \times 8$ ). Celkově tedy 11 bitů, přičemž některé koeficienty vychází záporně. Takto se ale bitová rychlost nesníží a je potřeba upravit velikost frekvenčních koeficientů kvantováním, tj. dělením čísla obsaženými v kvantizační matici [1].

V kvantizačních maticích ( $8 \times 8$  u MPEG-2) jsou uvedena vždy na odpovídající pozici osmibitová čísla, jimiž se dělí příslušný koeficient DCT, viz rovnice 3. Tato čísla se obvykle zvětšují směrem k vyšším prostorovým kmitočtům v souladu se skutečností, že lidské oko je na ně méně citlivé a mohou tedy být kvantovány hruběji. Koeficienty  $c_{u,v}$  transformace DCT se dělí různými celými čísly  $q_{u,v}$  a vznikají tak nové frekvenční koeficienty, které se zaokrouhlují na celé číslo [1].

$$s_{u,v} = \frac{c_{u,v}}{q_{u,v}}, \quad (2)$$

kde

$s_{u,v}$  je matice kvantovaných koeficientů,

$c_{u,v}$  je matice koeficientů DCT,

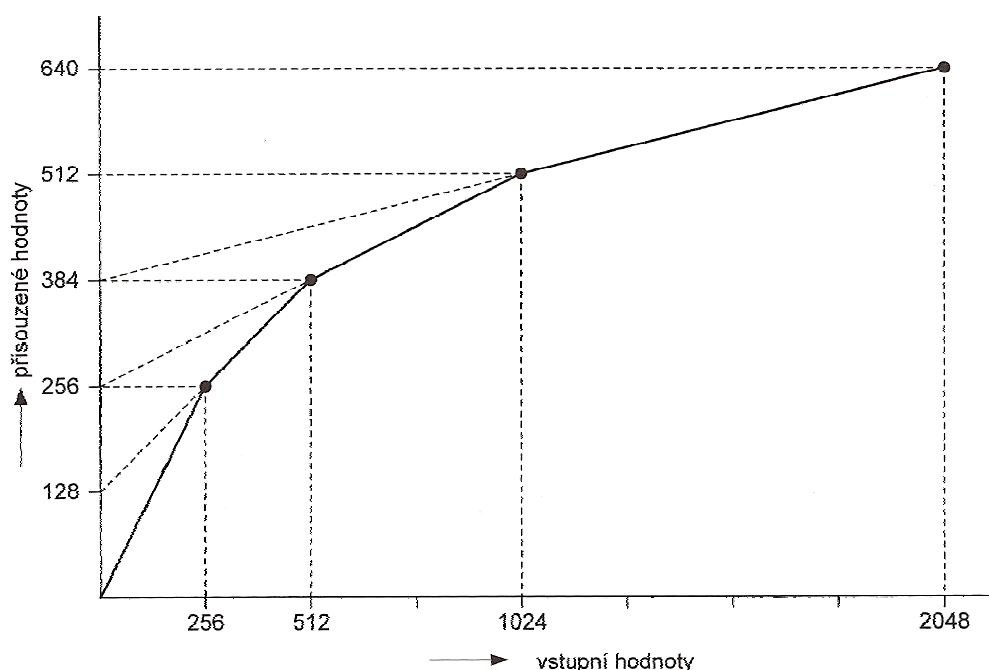
$q_{u,v}$  je kvantizační matice

$$q_{u,v} = \begin{bmatrix} 8 & 17 & 18 & 19 & 21 & 23 & 25 & 27 \\ 17 & 18 & 19 & 21 & 23 & 25 & 27 & 28 \\ 20 & 21 & 22 & 23 & 24 & 26 & 28 & 30 \\ 21 & 22 & 23 & 24 & 26 & 28 & 30 & 32 \\ 22 & 23 & 24 & 26 & 28 & 30 & 32 & 35 \\ 23 & 24 & 26 & 28 & 30 & 32 & 35 & 38 \\ 25 & 26 & 28 & 30 & 32 & 35 & 38 & 41 \\ 27 & 28 & 30 & 32 & 35 & 38 & 41 & 45 \end{bmatrix} \quad (3)$$



Po uvedeném vážení koeficientů DCT kvantizační maticí se provádí konečná kvantizace s tím, že je možné rozšířit tzv. mrtvou zónu kolem nuly za účelem vynulování maximálního počtu koeficientů DCT. Protože malé odchylky kolem nuly jsou způsobovány také šumem, jejich potlačení obvykle zlepšuje subjektivní kvalitu obrazu [1], [8].

Standard MPEG-2 používá dva typy standardních kvantizačních matic, v podstatě ale může kodér vytvořit libovolnou kvantizační matici a spolu se signálem ji přenést do dekodéru. U MPEG-2 se také frekvenční koeficienty kvantují nelineárně, viz obr. 4. Kvantizační tabulky pro jasový a chrominanční signál jsou různé. U většiny koeficientů chrominančního signálu, především u vyšších koeficientů, jsou dovoleny větší hodnoty činitelů kvantizační matice  $q_{u,v}$  [1], [8].



Obr. 4. Nelineární kvantování frekvenčních koeficientů [1].

## 2.3 Mezisnímkové kódování (Inter-frame)

Je založeno na skutečnosti, že následující snímky jsou více či méně podobné snímkům přecházejícím. Není třeba tedy komprimovat celé snímky, ale pouze jejich rozdíl (předpověď mezi snímky), který se vytváří diferenční pulsní kódovou modulací DPCM. Jedná se o redukci redundance v časové oblasti, tedy o bezztrátové kódování. Rozdíly v hodnotách vzorků dvou po sobě jdoucích snímků se vytváří na úrovni jednotlivých makrobloků. Spolu s transformačním ztrátovým kódováním tvoří tzv. hybridní komprimační kódování [1].

### 2.3.1 Druhy snímků a jejich předpověď

Klasický model MPEG-2 je založen na třech typech snímků:

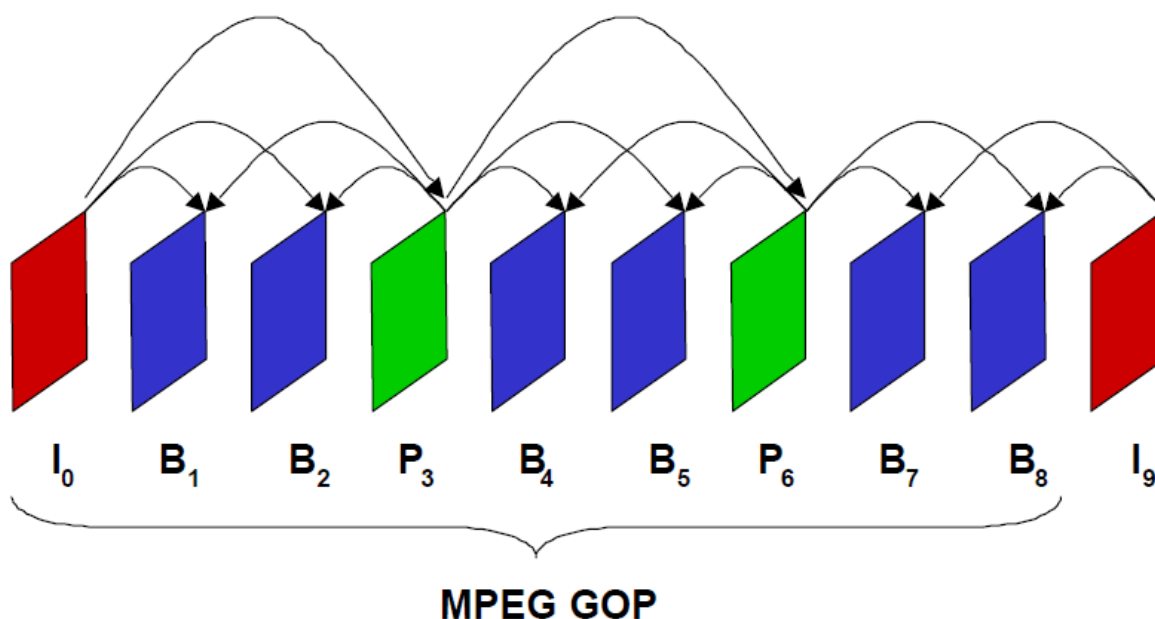
*Snímek typu I (intra frame).* Jedná se o tzv. referenční snímek, k jeho komprimaci, respektive dekomprimaci není potřeba žádného jiného snímku. Snímek se zpracovává přímo diskrétní kosinovou transformací DCT. Slouží jako předpověď pro snímky typu P a B.

*Snímek typu P (forward or backward prediction).* Pro kódování (dekódování) potřebuje jeden předcházející snímek typu I nebo P v případě, že se jedná o předpověď dopřednou (forward) a naopak následující snímek, opět typu I nebo P, pro předpověď zpětnou (backward). Při jednosměrné předpovědi, ať už zpětné nebo dopředné se bitová rychlost sníží asi dvakrát.

*Snímek typu B (forward and backward prediction – bidirectional).* Až osminásobné snížení bitové rychlosti se dosáhne obousměrnou předpovědí. Pro kódování a dekódování snímku typu B je potřeba předchozího a následujícího snímku, a to typu I nebo P. Kódovaný rozdíl je vytvořen jako rozdíl právě kódovaného snímku a průměru předchozího snímku a snímku následujícího [1], [7].

### 2.3.2 Skupina snímků (Group of Pictures)

Struktura GOP, viz obr. 5, obsahuje snímky typu P, B a jeden snímek typu I, který je na začátku GOP. Ostatní snímky jsou dopočítávány na základě předpovědi. Strukturu lze popsat dvěma parametry: N, což je počet snímků v GOP, respektive délka GOP a pro MPEG-2 je hodnota délky 12, a M, což je vzdálenost snímků P. Struktura GOP je IBBPBBPBBPBB. Snímek typu I je referenční snímek a nevyžaduje žádné dodatečné informace k jeho dekódování. Případné chyby ve struktuře GOP jsou korigovány dalším snímkem typu I.



Obr. 5. Skupina snímků GOP [1].

### 2.3.3 Vektor pohybu (Motion vector)

Při použití pouze diferenční pulzní kódové modulaci DPCM se vytvoří rozdíl v hodnotách vzorků na sobě odpovídajících místech – makroblocích – sousedících snímků. Tím dojde ke snížení potřebného počtu bitů. Zvýšení úspory bitů dosáhneme použitím tzv. vektorů pohybu [1].

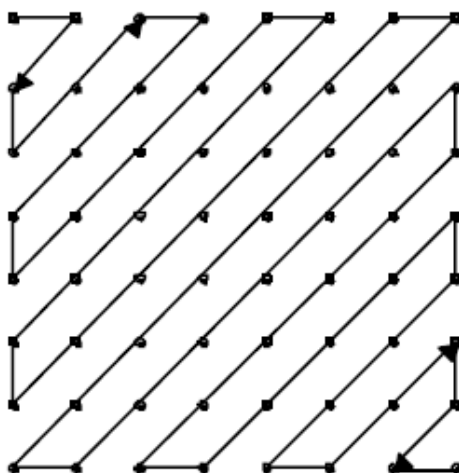
Princip spočívá v přenášení informace o pohybu objektů za pomoci vektorů pohybu a rozdílů příslušných makrobloků namísto přenosu rozdílů hodnot makrobloků, které si odpovídají pozicí. Vyhledávání je hardwarově náročné a proto se vyhledává na úrovni makrobloků, tj. při zpracování makrobloku kódovaného snímku se vyhledává v předchozím či následujícím snímku makroblok se stejným obrazovým obsahem, jehož pozice je určena pohybovým vektorem o souřadnicích  $x$ ,  $y$ . Hledání probíhá současně pro jasový i oba chrominanční signály, všem signálům pak náleží společný vektor pohybu [1].

Posunutí může být ale jen do maximální vzdálenosti 64 bodů (pro MPEG-2). Tomuto vektoru se říká vektor pohybu (motion vector) a metodě odhad pohybu (motion estimation). Je ale zřejmé, že v reálném světě není pohyb vždy tak jednoznačný, objekt se mohl otočit nebo zmenšit. Proto je ještě proveden výpočet rozdílu mezi původním a výsledným obrazem a tento je teprve pomocí DCT zakódován - tomuto principu se říká kompenzace pohybu (motion compensation). Rozdílová složka obsahuje většinou jen málo nenulových hodnot a tak se snadno zakóduje. Pokud by chyba byla příliš velká nebo nebyl nalezen vektor pohybu, kodér se může rozhodnout nekódovat rozdíl, ale zakódovat původní blok jako u I snímku. Ne všechny bloky tedy musí být pohybové [1], [7].

## 2.4 Entropické kódování

Entropie udává minimální počet bitů, který je potřeba pro vyjádření hodnoty vzorku v přenosu. Přiřazení bitů závisí na tom, s jakou pravděpodobností se hodnota vzorku vyskytuje v přenosu. Hodnotám s vyšší pravděpodobností výskytu se přiřazují slova, která jsou vyjádřena menším počtem bitů a naopak hodnotám, které se vyskytují málo, jsou přiřazena slova vyjádřena větším počtem bitů. Entropické kódování se také označuje jako statické [1].

Soustava MPEG 2 používá Huffmanovo kódování s proměnnou délkou slova (VLC – Variable Length Coding). Před samotným kódováním je potřeba přečíst matici frekvenčních koeficientů po kvantování tzv. cik-cak způsobem, viz obr. 6. Jedná se o čtení střídavých koeficientů AC podle úhlopříčky od levého horního rohu do pravého dolního rohu matice. Stejnosměrný koeficient DC se přenáší samostatně. Tento způsob je výhodný, neboť hodnoty koeficientů se zmenšují stejným směrem. Výstupem čtení je sériový tok dat, u něhož jsou od jistého koeficientu samé nuly [1].



Obr. 6. Čtení „cik-cak“ kvantovaných koeficientů v matici 8x8 [3].

Koeficienty se nekódují podle četnosti v bloku, ale zavádí se skupiny, které jsou složeny z hodnoty koeficientu a počtu předcházejících nul. Skupina je potom dána dvěma symboly. První symbol v sobě nese informaci o počtu nul a počtu bitů potřebných pro kódování frekvenčního koeficientu. Druhý symbol v přenášené skupině vyjadřuje hodnotu v binární formě. Bitové vyjádření prvního symbolu je dáno tabulkami ověřenými s určitou přesností v praxi [1].

## 2.5 Přehled profilů a úrovní standardu MPEG-2

MPEG profily, viz tab. 1, jsou specifická provedení nebo nastavení potřebných protokolů a opatření, která umožní poskytování funkcí a služeb pro konkrétní použití MPEG. Rozsah těchto aplikací se pohybuje od poskytování standardních služeb televizního vysílání přes vysílací systém pro poskytování služeb videa u mobilních sítí.

Profily jsou vytvořeny pro specifické aplikace a typy médií. Neustále jsou požadovány a vytvářeny nové profily. Vzhledem k tomu, že typy aplikací, které využívají MPEG přenos, se mohou výrazně lišit, má MPEG řadu profilů.

Úrovně jsou rozsahem možností, které profil MPEG může nabídnout. Použití úrovní umožňuje videozáznamům definovat jejich parametry a tím i jejich výslednou kvalitu, jako je velikost paměti, snímkování, rozlišení a maximální přenosová bitová rychlost. Úrovně se pohybují od nízké rozlišovací schopnosti po velmi vysokou rozlišovací schopnost [1], [6].

|                    | Jednoduchý profil         | Hlavní profil               | Odstupňovaný profil podle S/R  | Prostorově odstupňovaný profil  | Vysoký profil                                       |
|--------------------|---------------------------|-----------------------------|--------------------------------|---------------------------------|---|
| Nízká úroveň       |                           | 352 x 288 bodů<br>4 Mb/s    | 352 x 288 bodů<br>4 (3) Mb/s   |                                 |   |
| Hlavní úroveň      | 720 x 576 bodů<br>15 Mb/s | 720 x 576 bodů<br>15 Mb/s   | 720 x 576 bodů<br>15 (10) Mb/s |                                 | 720 x 576 bodů<br>(352 x 288)<br>20 (15,4) Mb/s     |
| Vysoká úroveň 1440 |                           | 1440 x 1152 bodů<br>60 Mb/s |                                | 1440 x 1152 bodů<br>(720 x 576) | 1440 x 1152 bodů<br>(720 x 576)<br>80 (60,20) Mb/s  |
| Vysoká úroveň      |                           | 1920 x 1152 bodů<br>80 Mb/s |                                |                                 | 1920 x 1152 bodů<br>(960 x 576)<br>100 (80,25) Mb/s |

Tab. 1. Profily a úrovně standardu MPEG-2 [1].

## 3 MPEG-4 Part 10 (H.264/AVC)

H.264 je standardem pro kompresi videa, proces konverze digitálního videa do formátu, který zabírá méně kapacity, když je uložen nebo při přenosu. Komprese videa (nebo kódování videa) je základní technologie pro aplikace, jako je digitální televize, DVD, mobilní televize, videokonference a internetové streamování videa. Standardizace komprese videa umožňuje, aby produkty od různých výrobců (např. kodéry, dekodéry a paměťová média) vzájemně spolupracovaly [9].

Standard H.264/AVC byl poprvé publikován v roce 2003. Vychází z koncepce starších standardů, jako MPEG-2 a MPEG-4 Visual, nabízí nástroje pro lepší účinnost komprese (tj. vyšší kvalita komprimovaného videa) a větší pružnosti při kompresi, ukládání a přenosu videa [9].

### 3.1 Blokové zapojení kodéru

Stejně jako kodér MPEG-2 je i kodér, viz obr. 7, MPEG-4 AVC rozdělen na části. Pro předpověď mezi snímky to jsou bloky:

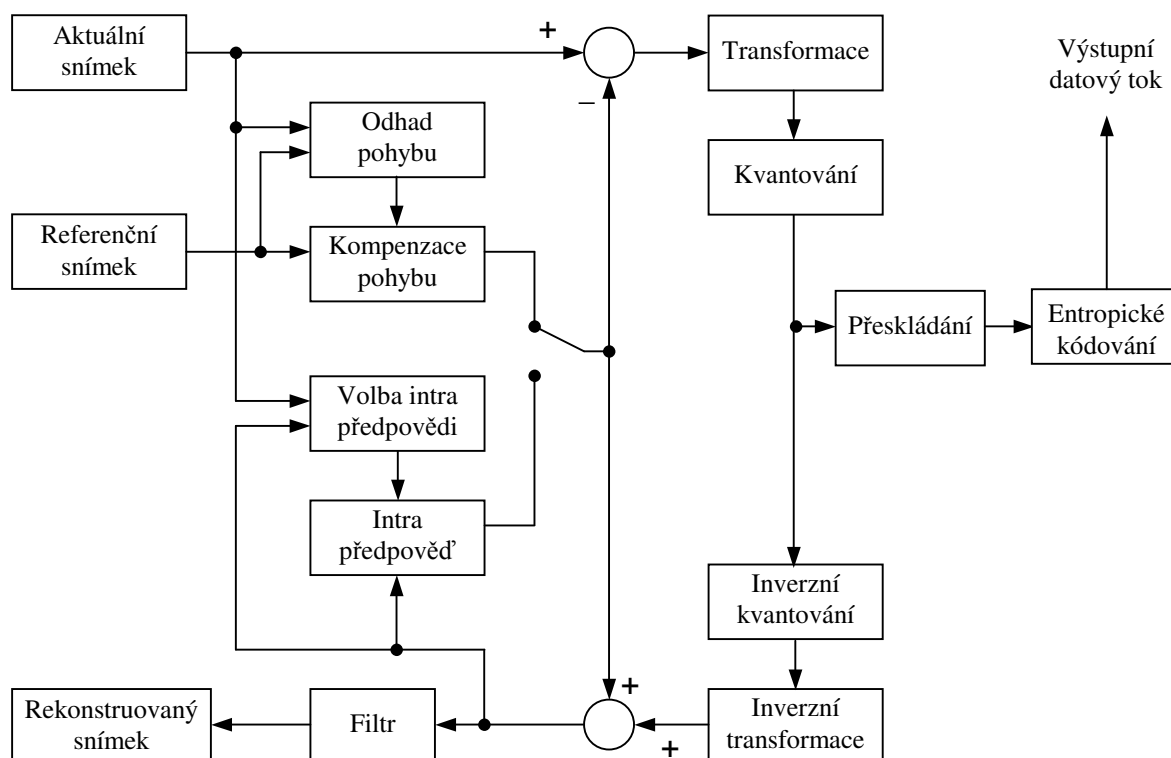
**Odhad pohybu** – Vytvoření pohybových vektorů na základě porovnání aktuálního a referenčního snímku.

**Kompenzace pohybu** – Aplikace pohybových vektorů na referenční snímek.

Pro předpověď uvnitř snímku potom:

**Volba intra předpovědi** – Volba předpovědi na základě aktuálního a rekonstruovaného (nefiltrovaného) snímku.

**Intra předpověď** – Aplikace předpovědi na rekonstruovaný (nefiltrovaný) snímek.



Obr. 7. Blokové zapojení kodéru MPEG-4 AVC [9].

Další bloky:

**Transformace** – Celočíselná DCT, Hadamardova transformace.

**Kvantování** - Dělení matice frekvenčních koeficientů kvantizační maticí.

**Inverzí kvantování, inverzní transformace** – Rekonstrukce rozdílového snímku, zpětné násobení kvantované matice a zpětná transformace  $DCT^{-1}$ .

**Entropické kódování** – „Cik cak“ čtení frekvenčních koeficientů a následné kódování.

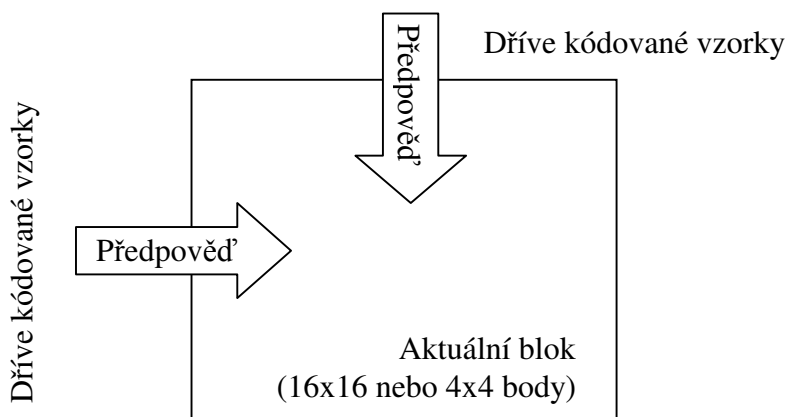
**Přeskládání** – Prohození rozdílových snímků pro správné dekódování na straně přijímače.

**Filtr** – Rekonstrukční filtr vyhlazuje ostré přechody mezi pixely.

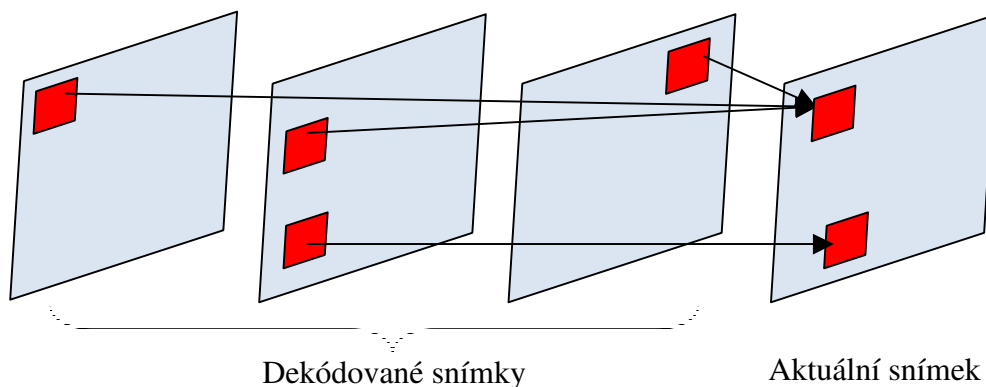
## 3.2 Předpověď (Prediction)

Kodér zpracovává videosnímek po makroblocích (16x16 obrazových bodů). Tvoří předpověď makrobloku na základě dříve kódovaných data, buď z aktuálního snímku (v rámci intra predikce) nebo z jiných snímků, které již byly dříve kódovány (inter predikce). Kodér odečítá předpověď od současného makrobloku a vytváří rozdíl. Nalezení vhodné předpovědi je popisováno jako motion estimation (odhad pohybu). Odečtení předpovědi ze současného makrobloku je označováno jako motion compensation (kompenzace pohybu) [9].

Predikční metody, které jsou H.264 podporovány jsou pružnější, než u předchozích norem, umožňuje přesné předpovědi a tudíž efektivní kompresi videa. V rámci intra predikce se používají velikosti bloků 16x16 a 4x4 pro předpověď makrobloku z okolních, viz obr. 8, dříve kódovaných obrazových bodů ve stejném snímku. Inter predikce využívá celou řadu velikostí bloků (od 16x16 po 4x4) pro předpověď bodů v aktuálním snímku z podobných oblastí u předešle kódovaných snímků, viz obr. 9 [9].



Obr. 8. Předpověď uvnitř snímku [9].



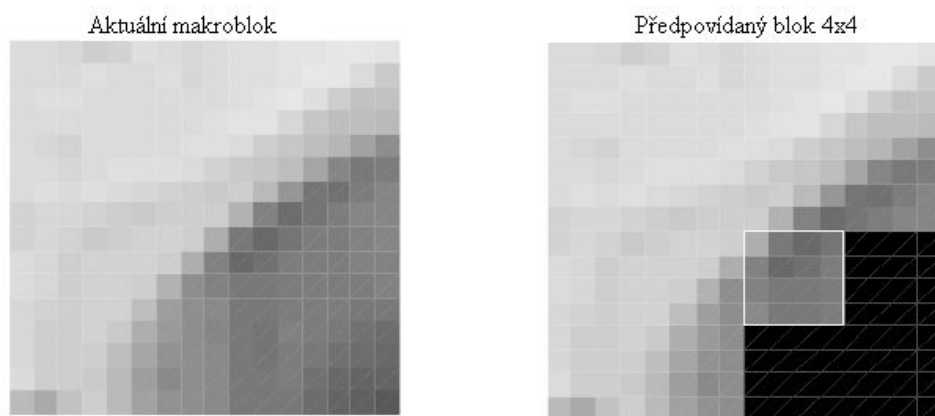
Obr. 9. Předpověď mezi snímky [9].

### 3.2.1 Přepověď uvnitř snímku (Intra prediction)

Je-li blok nebo makroblok kódován v intra módu je předpověď vytvořena na základě dříve kódovaných a rekonstruovaných (ale nefiltrovaných) blocích. Tato předpověď bloku P se odečte od stávajícího bloku před kódováním. Pro jasové (luminanční) vzorky může být blok P tvořen pro sub-makroblok (4x4 obrazové body) nebo pro makroblok (16x16 obrazových bodů). Existuje celkem 9 volitelných módů předpovědi pro jasový sub-makroblok (4x4) a 4 volitelné módy pro jasový makroblok (16x16). A potom jeden mód, který je vždy použit na každý chrominanční blok (4x4) [9].

#### 3.2.1.1 4x4 jasové módy předpovědi

Obr. 10 ukazuje jasový makroblok a předpovídaný 4x4 jasový blok. Vzorky nahoře a vlevo byly dříve zakódovány a rekonstruovány a jsou proto k dispozici v kodéru a dekodéru jako reference pro vytvoření předpovědi. Předpověď bloku P se počítá na základě vzorků označených A až M, viz obr. 11 [9].



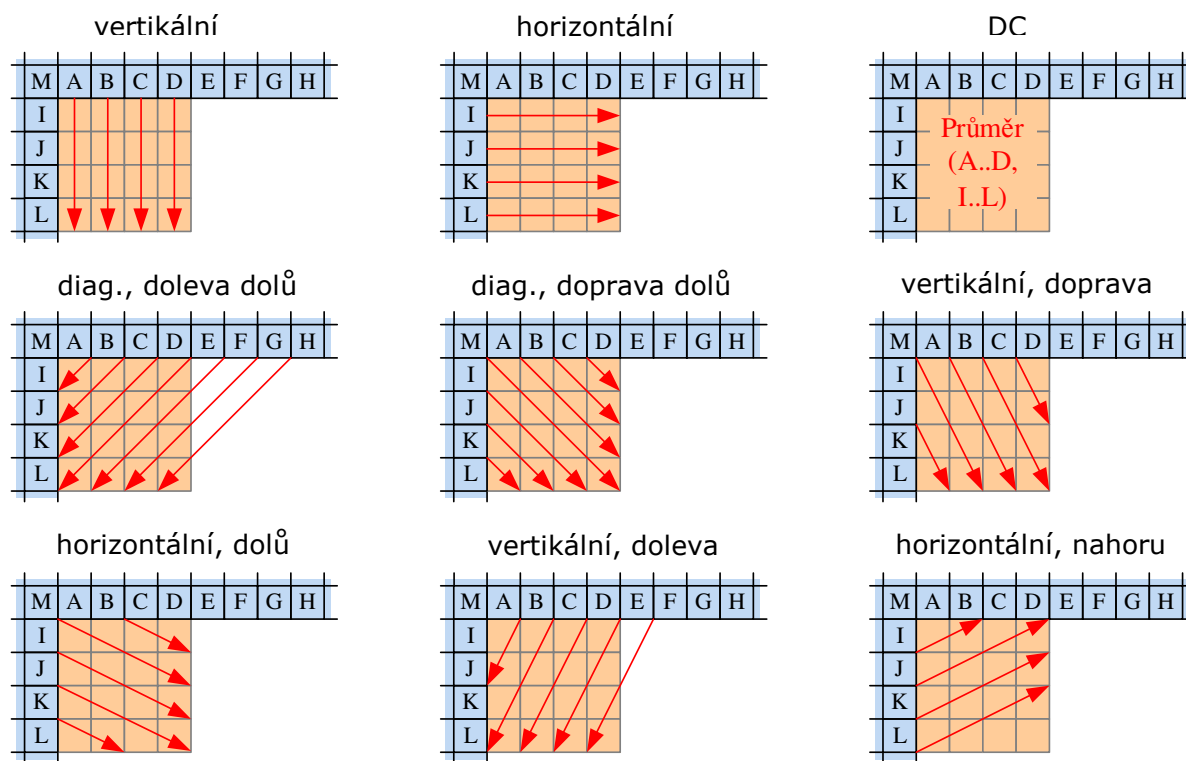
Obr. 10. Předpověď jasového bloku 4x4 [9].

V některých případech nejsou v současném proužku všechny vzorky A-M k dispozici. Z důvodu zachování nezávislého dekódování proužků jsou pro předpověď k dispozici pouze vzorky v současném proužku. DC předpověď (mód 0) se mění v závislosti na tom, které vzorky A-M jsou k dispozici. Ostatní módy (1-8) lze použít, pouze pokud jsou k dispozici všechny požadované předpovídané vzorky (kromě případu, kdy vzorky E, F, G a H nejsou k dispozici, jejich hodnota je zkopírována ze vzorku D a jsou označeny jako dostupné) [9].

| M | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| I | a | b | c | d |   |   |   |   |
| J | e | f | g | h |   |   |   |   |
| K | i | j | k | l |   |   |   |   |
| L | m | n | o | p |   |   |   |   |

Obr. 11. Označení vzorků předpovědi [9].

Šipky u jednotlivých módů, viz obr. 12, naznačují směr předpovědi. U módů 3-8 jsou předpovídané vzorky počítány z váženého průměru předpovědi vzorků A-M. Kodér může vybrat mód předpovědi pro každý blok tak, aby byla minimální difference mezi blokem P a blokem, který má být zakódován [9].

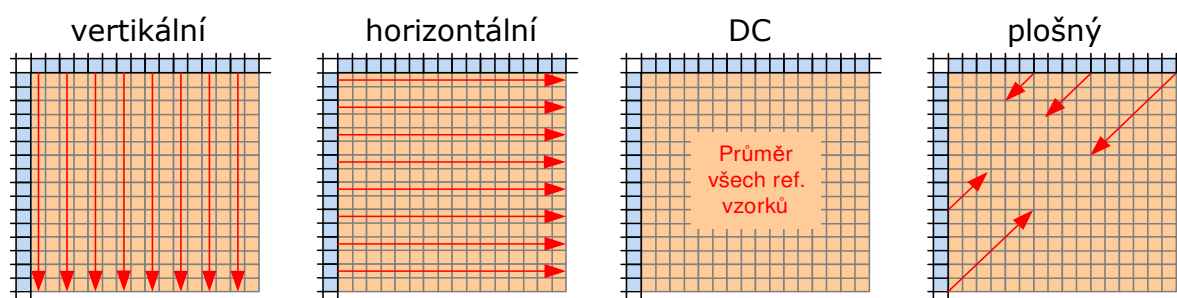


Obr. 12. 4x4 jasové módy předpovědi [10].

### 3.2.1.2 16x16 jasové módy předpovědi

Jako alternativu k 4x4 jasovým módům je 16x16 jasový mód pro předpověď celého makrobloku. K dispozici jsou 4 módy, viz obr. 13:

- Mód 0 (vertikální): extrapolace z horních vzorků (H).
- Mód 1 (horizontální): extrapolace z levých vzorků (V).
- Mód 2 (DC): průměr horních a levých vzorků (H + V).
- Mód 4 (Plošný): lineární "plošná" funkce mezi horními a levými vzorky H a V. Tento mód funguje dobře v oblastech s pozvolna se měnícím jasem.



Obr. 13. 16x16 jasové módy předpovědi [10].



### 3.2.1.3 8x8 chrominanční mód předpovědi

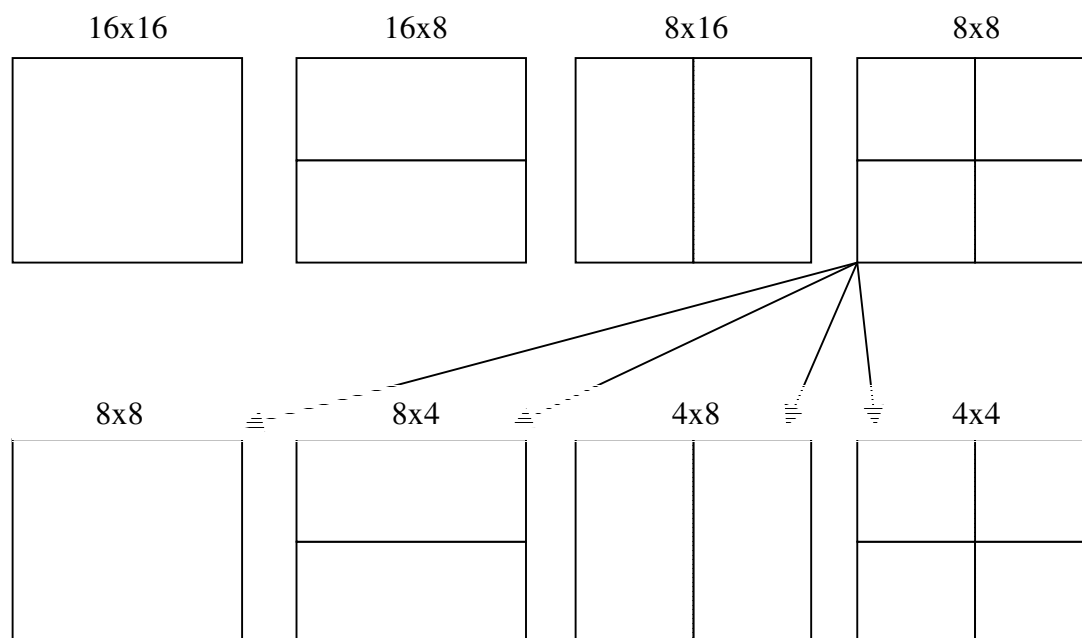
Každý 8x8 chrominanční makroblok je předpovídán z chrominančních vzorků nahoře a/nebo vlevo, které byly předtím zakódovány a rekonstruovány. Čtyři módy předpovědi jsou velmi podobné 16x16 jasovým módům. S výjimkou, že pořadí čísel módů jsou jiná. Na oba chrominanční bloky je použit stejný mód [9].

### 3.2.2 Přepověď mezi snímky (Inter prediction)

Inter předpověď vytváří model předpovědi z jednoho nebo více, dříve kódovaných snímků. Model je tvořen přesouváním vzorků v referenčním snímku či snímcích (motion compensated prediction – předpověď kompenzace pohybu). AVC kodek využívá blokovou kompenzaci pohybu, stejný princip využívají všechny hlavní kódovací standardy od H.261. Významné rozdíly od předchozích standardů zahrnují podporu pro rozsah velikosti bloků (až do 4x4) a pohybové vektory o rozlišení 1/4 pixelu pro jasové vzorky a 1/8 pixelu pro chrominanční vzorky [9].

#### 3.2.2.1 Stromová struktura pohybové kompenzace

AVC podporuje velikost bloků kompenzace pohybu od 16x16 do 4x4 jasových vzorků s mnoha variantami. Každý jasový makroblok (16x16 vzorků) může být rozdělen na 4 způsoby: 16x16, 16x8, 8x16 nebo 8x8. Jestliže je zvoleno rozdělení 8x8, tak každý ze čtyř 8x8 sub-bloků může být rozdělen na další 4 způsoby: 8x8, 8x4, 4x8 nebo 4x4. Tyto bloky a sub-bloky vedou k velkému počtu možných kombinací uvnitř každého makrobloku, viz obr. 14 [9].

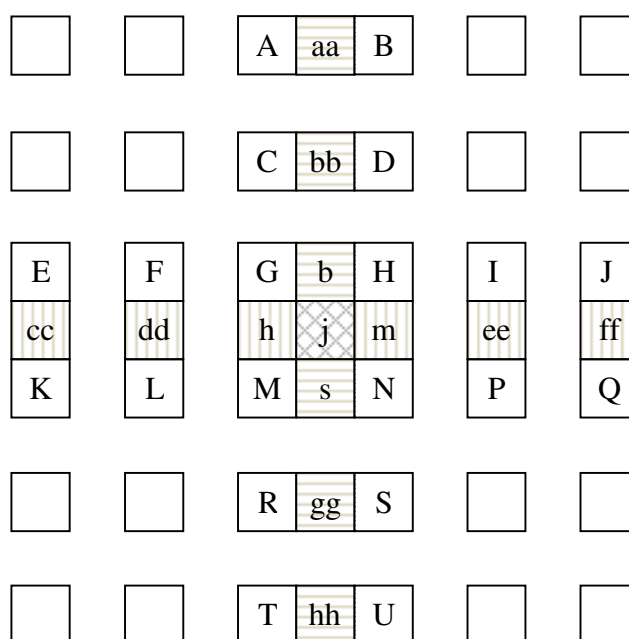


Obr. 14. Stromová struktura [9].

### 3.2.2.2 Vektor pohybu (Motion vector)

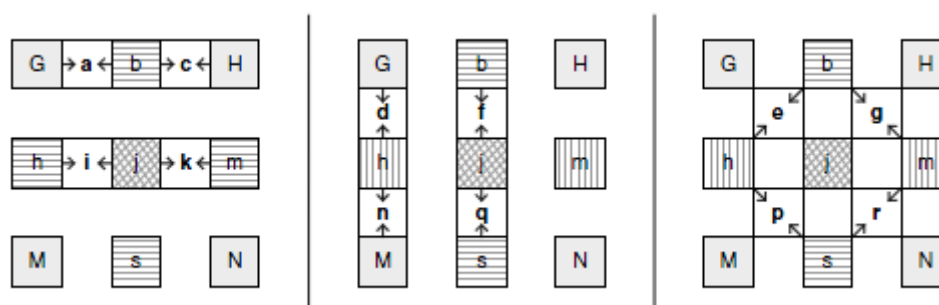
Každý blok v inter kódovaném makrobloku je předpovídaný z oblasti o stejné velikosti v referenčním obraze. Posunutí (pohybový vektor) mezi oběma oblastmi má rozlišení  $\frac{1}{4}$  pixelu pro jasovou složku. Jasové a chrominanční vzorky na pozici sub-pixelu, tj. mezi pixely, v referenčním obraze neexistují a proto je nutné je vytvořit pomocí interpolace z okolních vzorků obrazu [9].

Mezipixelová kompenzace pohybu může poskytnout podstatně lepší kompresi než kompenzace, která pracuje s rozlišením celých pixelů, ovšem na úkor zvýšené složitosti. U jasové složky jsou sub-pixelové vzorky v poloze  $\frac{1}{2}$  pixelu generovány jako první a jsou interpolovány ze sousedních vzorků na pozici celých pixelů za pomoci FIR filtru s vahami  $\frac{1}{32}$ ;  $-\frac{5}{32}$ ;  $\frac{5}{8}$ ;  $\frac{5}{8}$ ;  $-\frac{5}{32}$ ;  $\frac{1}{32}$ . To znamená, že hodnota vzorku v pozici  $\frac{1}{2}$  pixelu je dána váženým součtem šesti sousedních vzorků na celistvých pozicích pixelu, viz obr. 15 [9].



Obr. 15. Interpolace hodnot vzorků v poloze  $\frac{1}{2}$  pixelu [9].

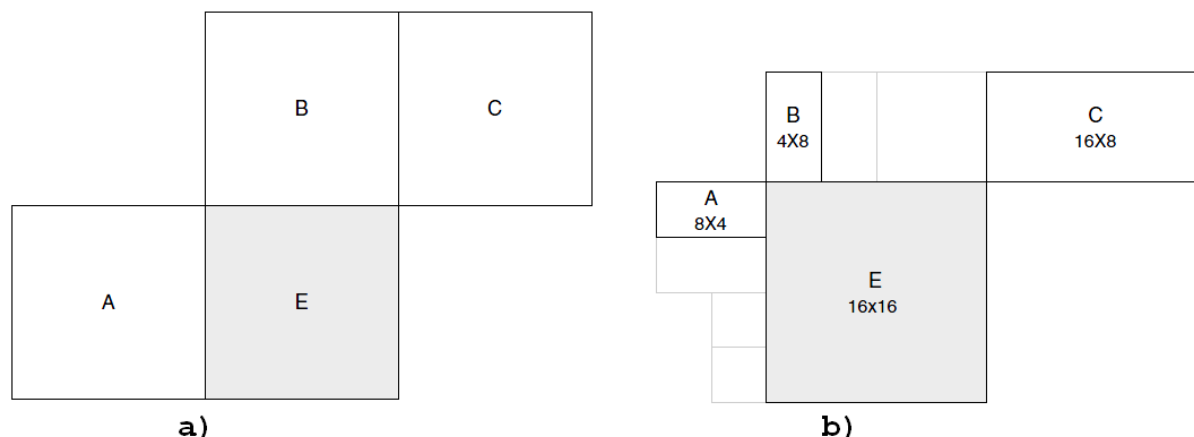
Jakmile jsou všechny vzorky v poloze  $\frac{1}{2}$  pixelu k dispozici, dochází za použití lineární interpolace k výpočtu (mezi dvěma hodnotami v poloze  $\frac{1}{2}$  pixelu nebo hodnotou na pozici celého pixelu a hodnotou v poloze  $\frac{1}{2}$  pixelu) hodnot vzorků v poloze  $\frac{1}{4}$  pixelu, viz obr. 16 [9].



Obr. 16. Interpolace hodnot vzorků v poloze  $\frac{1}{4}$  pixelu [9].

### 3.2.2.3 Předpověď vektoru pohybu

Kódování vektoru pohybu pro každý blok může zabrat značný počet bitů, zejména v případě malých velikostí bloků. Vektory pohybu sousedních bloků jsou často vysoce korelovány, proto je vektor pohybu předpovídaný z okolních, dříve kódovaných bloků. Rozdíl mezi aktuálním a předvídaným vektorem pohybu je kódován a přenášen. Metody vytváření předpovědi vektoru pohybu závisí na velikosti bloků a na dostupnosti okolních vektorů. Základem předpovědi je střední hodnota pohybových vektorů bloků nebo sub-bloků makrobloku, které jsou okolo, respektive nahoře, vlevo a diagonálně nahoře vpravo vůči pozici aktuálního bloku či sub-bloku, viz obr. 17 [9].



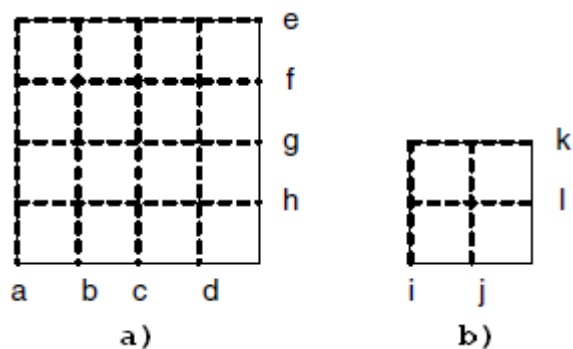
Obr. 17. Předpověď bloku E ze: a) stejné velikosti bloků b) rozdílné velikosti bloků [9].

## 3.3 Rekonstrukční filtr (Deblocking Filter)

Filtr je aplikován na všechny dekódované makrobloky s cílem omezit zkreslení blokovými artefakty. Filtr je aplikován v kodéru po inverzní transformaci (před rekonstrukcí a uložením makrobloku do paměti pro budoucí předpověď) a v dekóderu (před rekonstrukcí a zobrazením makrobloku). Filtr má dvě výhody: (1) hrany bloku se vyhladí, zlepší vzhled dekódovaného obrázku (zejména při vyšším kompresním poměru) a (2) filtrované makrobloky se používají pro předpověď pohybové kompenzace budoucích snímků, výsledkem je menší difference po předpovědi. (Poznámka: Intra kódované makrobloky jsou filtrovány, ale předpověď se v rámci intra módu provádí pomocí nefiltrovaných rekonstruovaných makrobloků) [9].

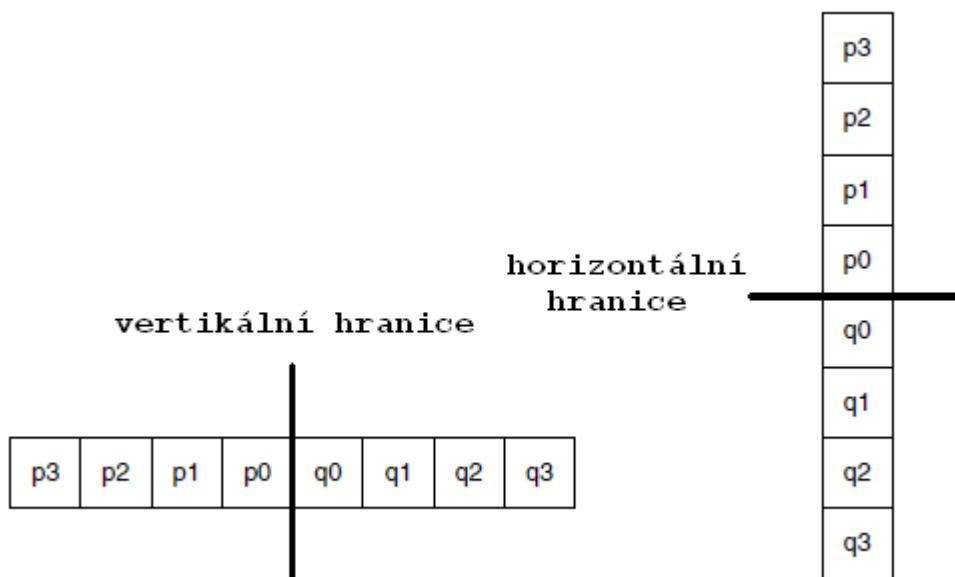
Filtrování je aplikováno na svislé nebo vodorovné okraje bloků, o velikosti 4x4 obrazových bodů v makrobloku, v následujícím pořadí, viz obr. 18 [9]:

1. Filtrování 4 vertikálních hranic jasové složky (v pořadí a, b, c, d).
2. Filtrování 4 horizontálních hranic jasové složky (v pořadí e, f, g, h).
3. Filtrování 2 vertikálních hranic barevné složky (i, j).
4. Filtrování 2 horizontálních hranic barevné složky (k, l).



Obr. 18. Pořadí filtrování v makrobloku a) jasové složky b) barevné složky [9].

Každé filtrování ovlivňuje až tři vzorky na každé straně hranice, viz obr. 19. Míra filtrování závisí na aktuálním kvantování, způsobu kódování sousedních bloků a na rozdílu hodnot vzorků obrazu na hranici [9].

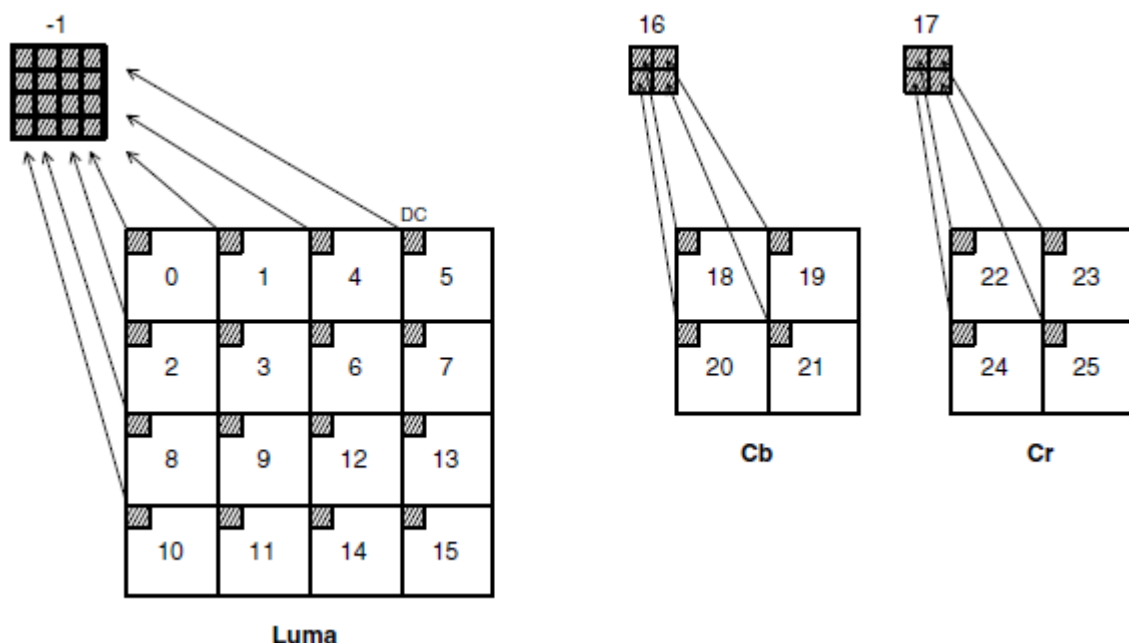


Obr. 19. Vzorky v blízkosti svislé a vodorovné hranice [9].

### 3.4 Transformační kódování (Transform)

H.264 používá tři transformace v závislosti na typu rozdílových dat, která se mají kódovat: Hadamardova transformace pro blok 4x4 jasových DC koeficientů (pouze u kódování makrobloku v módu 16x16), Hadamardova transformace pro blok 2x2 DC koeficientů barevných složek (v každém makrobloku) a transformace založená na DCT pro všechny ostatní bloky 4x4 rozdílových dat [9].

Je-li makroblok, viz obr. 20, kódován v módu 16x16, pak jsou data v rámci makrobloku přenášeny v pořadí, kdy první se přenáší blok označen -1, který obsahuje DC koeficienty každého bloku 4x4 jasové složky. Dále se přenáší rozdílové bloky 0-15, ovšem bez DC koeficientů. Následují bloky 16 a 17 obsahující matici 2x2 DC koeficientů barevných složek Cb a Cr a nakonec jsou zasílány rozdílové bloky barevných složek 18-25, také bez DC koeficientů [9].



Obr. 20. Transformační kódování MPEG-4 AVC [9].

### 3.5 Přehled profilů

**Profil Baseline** – Primárně určený pro aplikace, které vyžadují odolnost vůči ztrátě dat. Využívaný pro mobilní aplikace, videokonference, viz tab. 2.

**Profil Main** – Profil určený pro standardní televizní vysílání DVB.

**Profil High** - Profil je určený primárně pro televizní vysílání, zejména DVB HDTV. Je také využíváný pro BluRay disky.

**Profil Extended** – Určený pro streamování videa, dosahuje vysoké komprese. Odolnost vůči ztrátě dat.

| Nástroj                      | Profil   |      |      |          |
|------------------------------|----------|------|------|----------|
|                              | Baseline | Main | High | Extended |
| I a P snímky                 | X        | X    | X    | X        |
| CAVLC kódování               | X        | X    | X    |          |
| CABAC kódování               |          | X    | X    |          |
| B snímky                     |          | X    | X    | X        |
| Kódování prokládaného videa  |          | X    | X    | X        |
| Flexibilní pořadí makrobloku | X        |      |      | X        |
| Volné pořadí snímků          | X        |      |      | X        |
| Redundantní snímky           | X        |      |      | X        |
| Rozdělení dat                |          |      |      | X        |
| SI a SP snímky               |          |      |      | X        |

Tab. 2. Přehled profilů a dostupných nástrojů AVC [9].

## 4 VC-1 (WMV9)

VC-1 kodek je navržen k dosažení vysoké kvality komprimovaného videa v rozsahu velmi nízkých až velmi vysokých bitových rychlostí. Kodek snadno zvládá rozlišení 1920x1080 pixelů v rozsahu 6 až 30 Mb/s pro HD video. VC-1 je schopen pracovat i s vyšším rozlišením jako 2048x1536 pixelů pro digitální kina, s maximální přenosovou rychlostí 135 Mb/s. Příklad velmi nízkých bitových rychlostí jsou modemové aplikace s rychlostí 10Kb/s a rozlišením 160x120 pixelů [19].

Základní funkčnost VC-1 zahrnuje systém blokové kompenzace pohybu a prostorové transformace podobný jako u jiných standardů pro kompresi videa. VC-1 však obsahuje řadu inovací a optimalizací, které jsou odlišné od základního schématu komprese, což vede k vysoké účinnosti a dosažení vysoké kvality. VC-1 Advanced Profil je také transportně a kontejnerově nezávislý. To poskytuje ještě větší flexibilitu pro výrobce zařízení a služeb [19].

### 4.1 Blokové zapojení kodéru

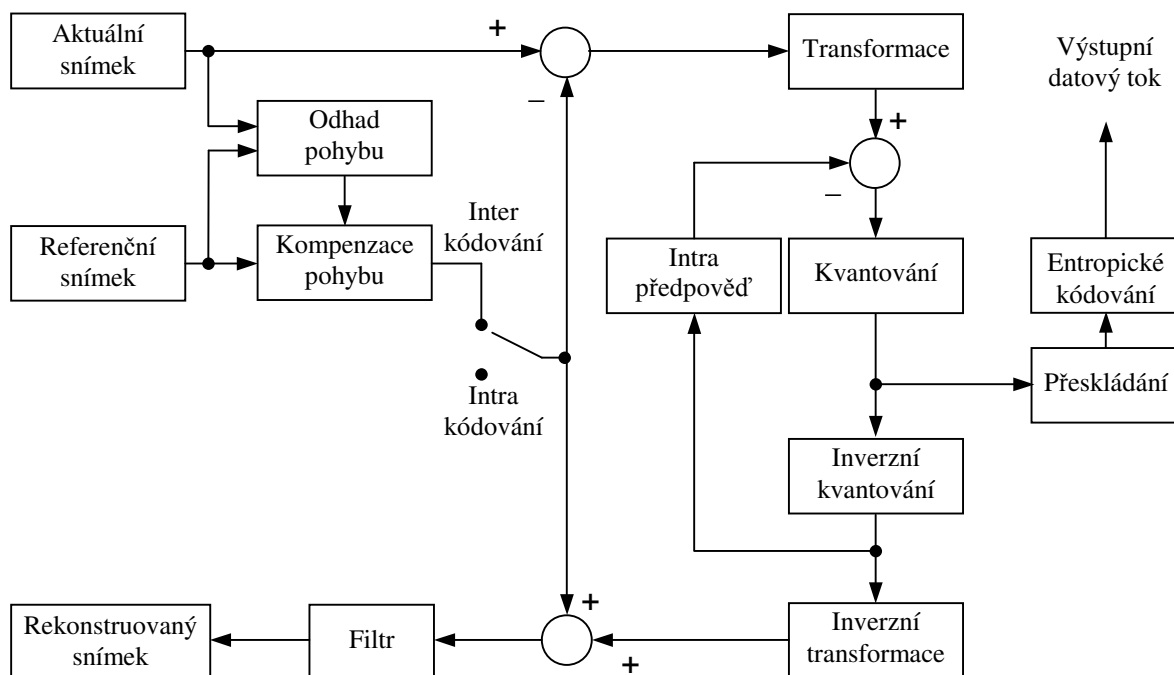
Kodér VC-1, viz obr. 21, je podobný jako kodér AVC s výjimkou různých bloků pro Intra předpověď. Pro předpověď mezi snímky slouží bloky:

**Odhad pohybu** – Vytvoření pohybových vektorů na základě porovnání aktuálního a referenčního snímku.

**Kompenzace pohybu** – Aplikace pohybových vektorů na referenční snímek.

Pro předpověď uvnitř snímku potom:

**Intra předpověď** – Předpověď frekvenčních koeficientů makrobloků a bloků ve frekvenční oblasti.



Obr. 21. Blokové zapojení kodéru VC-1 [20].

Další bloky:

**Transformace** – Celočíselná DCT.

**Kvantování** - Dělení matice frekvenčních koeficientů kvantizační maticí.

**Inverzí kvantování, inverzní transformace** – Rekonstrukce rozdílového snímku, zpětné násobení kvantované matice a zpětná transformace  $DCT^{-1}$ .

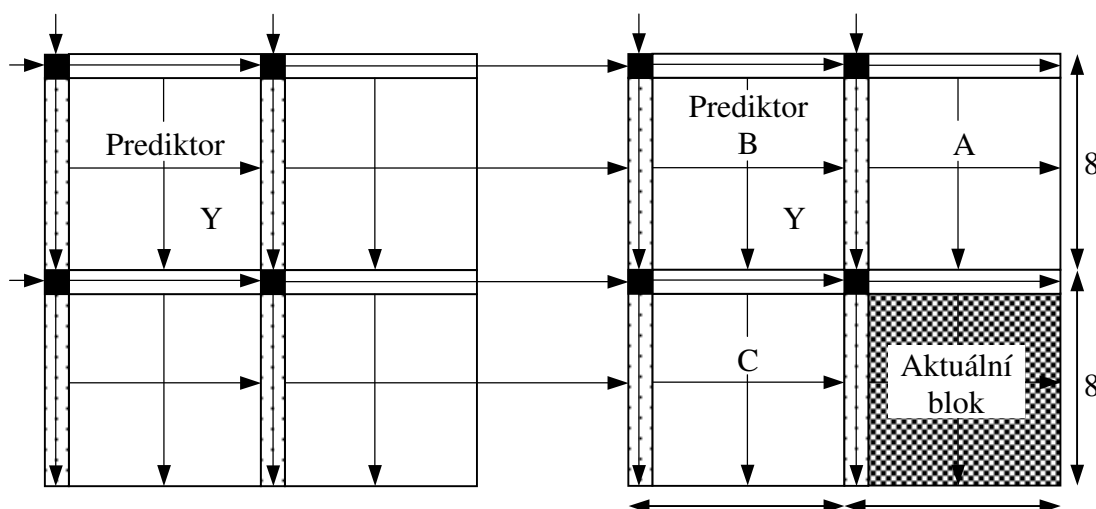
**Entropické kódování** – „Cik cak“ čtení frekvenčních koeficientů a následné kódování.

**Přeskládání** – Prohození rozdílových snímků pro správné dekódování na straně přijímače.

**Filtr** – Rekonstrukční filtr vyhlazuje ostré přechody mezi pixely.

## 4.2 Předpověď uvnitř snímku (Intra prediction)

VC-1 podporuje DC/AC předpověď ve frekvenční oblasti, jako je například u MPEG-4 part 2. DC předpověď je vždy povinná, zatímco AC předpověď je volitelná. Předpovědi pro jasovou složku a chrominanční složky jsou na sobě nezávislé. VC-1 zavádí Intra bloky 8x8 uvnitř Inter makrobloku, ten tak může mít v režimu se čtyřmi pohybovými vektory více jak jeden Intra blok 8x8. Intra makrobloky jsou u snímku I, volitelně pak u snímku P/B. [19].

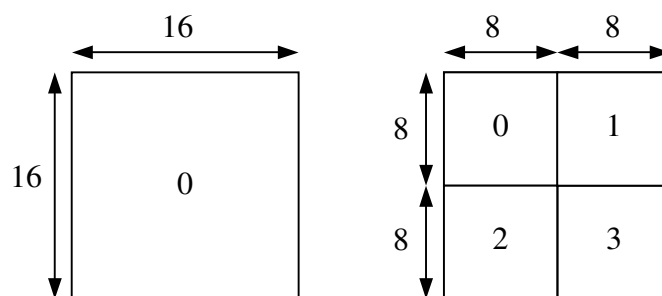


Obr. 22. Intra DC/AC předpověď [19].

Pro zakódování DC koeficientu aktuálního bloku, viz obr. 22, je vybrán horizontální směrový prediktor nebo vertikální horizontální prediktor. Pokud jsou prediktory bloku A a C dostupné, tak se směr predikce určí z bloku C, v případě, že absolutní hodnota rozdílu hodnot DC koeficientů bloků B a A je menší nebo rovna absolutní hodnotě rozdílu hodnot DC koeficientů bloků B a C. V opačném případě se směr predikce určí z bloku A. Pokud je dostupný pouze prediktor bloku C, tak je prediktor vybrán z něj. Předpověď AC koeficientů se provádí podobným způsobem pro první řádek nebo sloupec AC koeficientů předpovídaných ve směru stanoveném pro DC koeficient.

## 4.3 Předpověď mezi snímky (Inter prediction)

Velikost bloku pohybové kompenzace je 16x16 pro jasový blok. Ten lze rozdělit do čtyř bloků 8x8, viz obr. 23, pro individuální pohybovou kompenzaci. Velikost bloku 8x8 je velmi efektivní, není-li oblast pokrytí jednotná v pohybu textury jako například hranice objektu [19].



Obr. 23. Bloky pohybové kompenzace VC-1 [19].

### 4.3.2 Interpolace (Interpolation)

Pro neprokládané video je velikost bloku jasové složky odhadu pohybu a pohybové kompenzace 16x16 nebo 8x8. U prokládaného videa může být velikost poloviční. Kodér algoritmu VC-1 si pro interpolaci vybírá z několika interpolačních metod. Hlavní myšlenkou je upřednostnění metod pohybové kompenzace založených na kombinaci velikosti bloku pohybové kompenzace, interpolační metody a rozlišení vektoru pohybu. Podle typu zařízení volí kodér metodu pohybové kompenzace. VC-1 zavádí čtyři metody pohybové kompenzace, metody jsou seřazeny podle složitosti a kvality [19]:

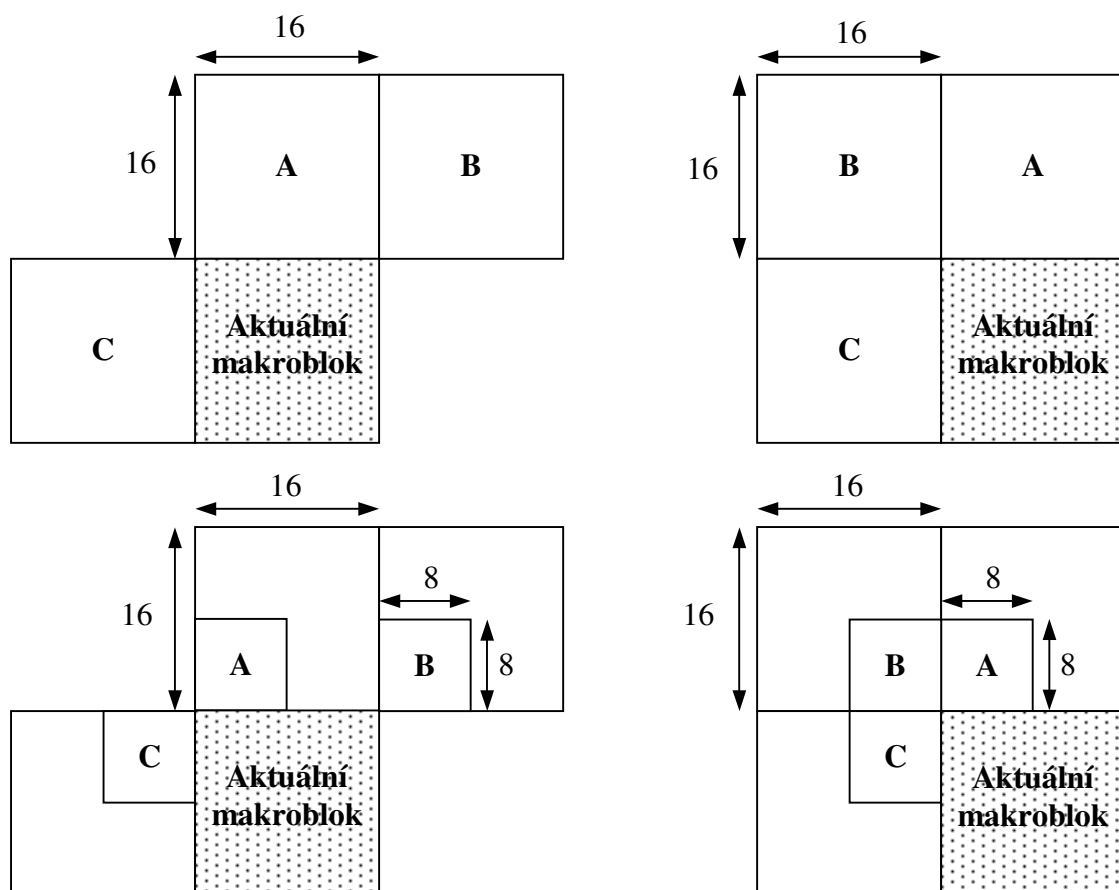
1. velikost bloku 16x16, 1 pohybový vektor, rozlišení  $\frac{1}{2}$  pixelu, bilineární interpolace
2. velikost bloku 16x16, 1 pohybový vektor, rozlišení  $\frac{1}{2}$  pixelu, bikubická interpolace
3. velikost bloku 16x16, 1 pohybový vektor, rozlišení  $\frac{1}{4}$  pixelu, bikubická interpolace
4. velikost bloků 8x8, 4 pohybové vektory, rozlišení  $\frac{1}{4}$  pixelu, bikubická interpolace

Interpolační metody pro bloky jasové složky a chrominančních složek jsou identické. Tyto metody jsou dvě – bilineární a bikubická. Pro bilienární metodu není důležitý sled operací, protože výsledek je definován jako výpočet ze čtyř nejbližších celých pixelů. U bikubické interpolace je sled operací zásadní, vertikální před horizontálními, neboť výsledek je definován výpočtem čtyř po sobě jdoucích celých pixelů [19].

### 4.3.3 Předpověď vektoru pohybu (Motion Vector Predictors)

Prediktor vektoru pohybu makrobloku je medián ze tří vektorů pohybu sousedních makrobloků nebo bloků. Horní dva vzory, viz obr. 24, jsou určeny pro velikost bloku 16x16 pohybové kompenzace, v obvyklém případě je pro volbu použit první vzor. Pokud je ovšem pozice aktuálního makrobloku poslední v řadě, makroblok B není k dispozici. V takovém případě je použit druhý vzor. Dolní dva vzory jsou určeny pro smíšenou velikost bloků pohybové kompenzace, přičemž aktuální makroblok je velikosti 16x16 a sousední bloky velikosti 8x8. Pro předpověď jsou vybrány nejbližší bloky 8x8 ze čtyř bloků každého makrobloku [19].

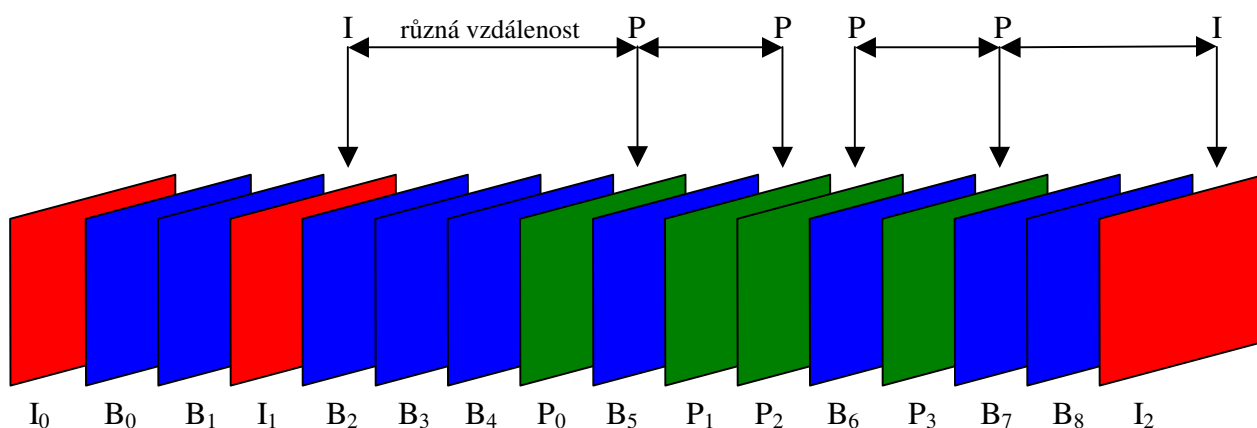




Obr. 24. Předpověď vektoru pohybu aktuálního makrobloku [19].

## 4.4 Skupina snímků (Group of Pictures)

U algoritmu VC-1 není stanovena pevná struktura GOP, viz obr. 25. Snímky I a P mohou být referenční snímky a výběr reference zcela závisí na volbě kodéru. Není zde stanoven žádný parametr  $N$  (počet snímků v GOP) a vzdálenost mezi dvěma referenčními snímky je proměnná. Maximální počet snímků B mezi dvěma referenčními snímky je 7 [19].



Obr. 25. Skupina snímků s volnou strukturou [19].

## 4.5 Transformační kódování (Transform)

Transformace jsou u VC-1 počítány v celočíselné aritmetice, tak dochází k eliminaci nepřesností po inverzní transformaci. VC-1 podporuje adaptivní velikost bloků: 8x8, 8x4, 4x8 a 4x4. Tato funkce přispívá ke zlepšení kvality kódování u videa, která obsahuje velmi objekty s jemnými detaily. Zaokrouhlení na nejbližší celá čísla podle vztahu [19]:

$$H = \text{round}\{\alpha H_{DCT}\}, \quad (4)$$

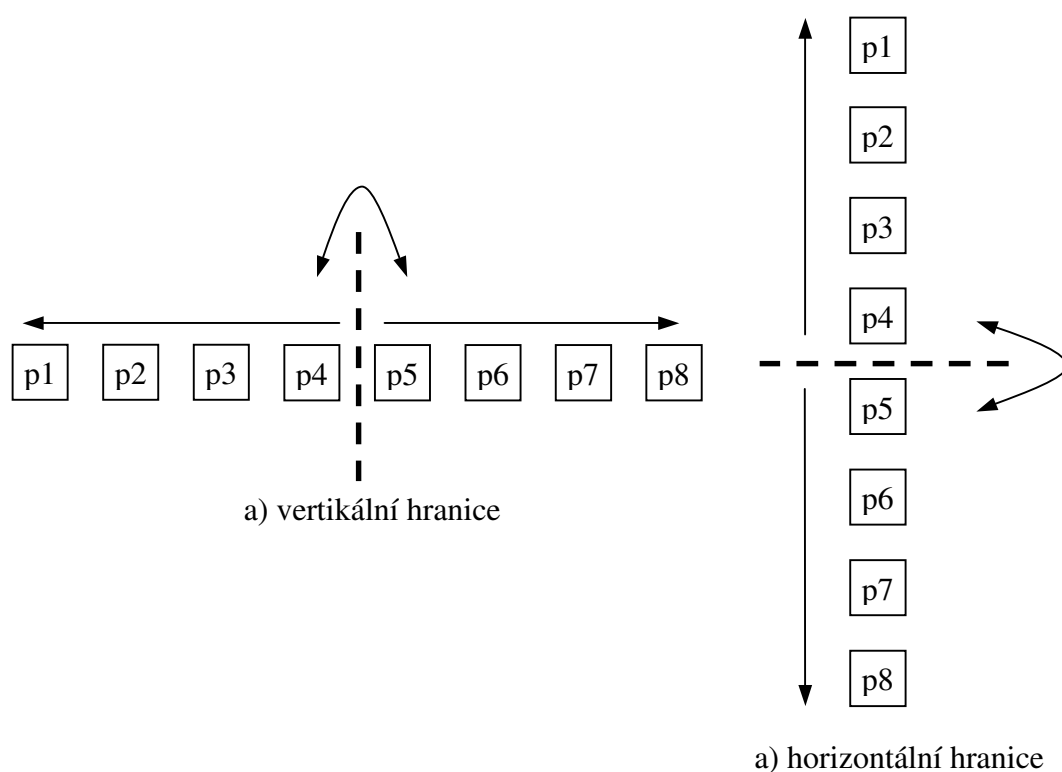
kde

$H_{DCT}$  je matice transformovaných koeficientů

## 4.6 Rekonstrukční filtr (Deblocking Filter)

Myšlenka rekonstrukčního filtru spočívá v aplikování filtrace na 8 pixelů rozdílných hodnot s hranicí uprostřed, viz obr. 26. Filtrace je prováděna u všech bloků 8x8 snímků I nebo B. Všechny horizontální hranice bloků jsou filtrovány jako první, pak následují vertikální hranice. U snímků P mohou být bloky kódovány Intra nebo Inter. Intra kódované bloky používají inverzní transformaci 8x8, zatímco Inter kódované bloky používají inverzní transformaci 8x8, 8x4, 4x8 nebo 4x4. Hranice mezi transformovanými bloky jsou filtrovány vždy, s výjimkou, kdy jsou transformované bloky na obou stranách hranice Inter kódovány, jejich vektory pohybu jsou totožné a všechny transformační koeficienty jsou nulové, v takovém případě k filtrování nedochází [19].

Stupeň filtrování je adaptivně volen podle určitých kritérií. U prvního stupně je filtrování aplikováno na každý třetí ze čtyř pixelů v bloku. V druhém stupni jsou filtrovány všechny pixely [19].



Obr. 26. Definice hranic pro filtrování [19].

## 4.7 Přehled profilů

**Profil Simple** – Zaměřuje se na internetové vysílání s malou bitovou rychlostí a na aplikace s nízkou komplexností jako jsou mobilní komunikace, média v osobních přehrávačích. Profil obsahuje dvě úrovně – nízká a střední úroveň.

**Profil Main** – Zaměřuje se na internetové aplikace s vysokou bitovou rychlostí – televizní vysílání, přehrávání filmů online. Profil obsahuje tři úrovně – nízká, střední a vysoká úroveň.

**Profil Advanced** – Určený pro vysílací aplikace jako je digitální televizní vysílání, HDTV. Je určen pro HD DVD a Blu-ray disky. Jako jediný podporuje prokládání. Profil obsahuje pět úrovní – L0, L1, L2, L3 a L4 [19].

Následující tabulka, viz tab. 3, obsahuje výpis nástrojů a jejich dostupnosti pro jednotlivé profily algoritmu VC-1.

| Nástroj  | Profil |      |          |
|--|--------|------|----------|
|  | Simple | Main | Advanced |
| Intra snímková předpověď                                 | X      | X    | X        |
| Transformace bloků proměnné velikosti                    | X      | X    | X        |
| 16-bitová transformace                                   | X      | X    | X        |
| Transformace s překrytím                                 | X      | X    | X        |
| 4 pohybové vektory na makroblok                          | X      | X    | X        |
| Kompenzace pohybu ¼ pixelu pro jasovou složku Y          | X      | X    | X        |
| Kompenzace pohybu ¼ pixelu pro barevné složky U, V       |        | X    | X        |
| Vyhlazovací filtr  |        | X    | X        |
| Dynamická změna rozlišení                                |        | X    | X        |
| Adaptivní kvantizace makrobloku                          |        | X    | X        |
| B snímky   |        | X    | X        |
| Kompenzace intenzity                                     |        | X    | X        |
| Nastavení rozsahu  |        | X    | X        |
| Kódování prokládaného videa                              |        |      | X        |
| Metadata pro zobrazení (kolorimetrie, poměr stran, atd.) |        |      | X        |

Tab. 3. Přehled profilů a dostupných nástrojů VC-1 [19].

## 5 Metriky pro měření kvality

PSNR (**P**eam **S**ignal-to-**N**oise **R**atio), neboli špičkový poměr signálu k šumu vyjadřuje poměr mezi maximální možnou energií signálu a energií šumu. PSNR se obvykle vyjadřuje v logaritmickém měřítku, jelikož spousta signálů má velmi široký dynamický rozsah. Hodnota PSNR se udává v decibelech a typické hodnoty pro kompresi videa jsou mezi 30dB až 50dB. Za přípustné hodnoty jsou pro bezdrátový přenos považovány hodnoty 20dB až 25dB. PSNR je definován rovnicí [12]:

$$\text{PSNR}_{\text{dB}} = 10 \cdot \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right) = 20 \cdot \log_{10} \left( \frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right), \quad (5)$$

kde

$\text{MAX}_I$  je maximální hodnota pixelu obrazu a

MSE (**M**ean **S**quared **E**rror) je střední kvadratická chyba. MSE se určí jako:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2, \quad (6)$$

kde

$I, K$  jsou černobílé obrazy a

$m, n$  jsou rozměry.

SSIM (**S**tructural **S**IMilarity) index je navržen s cílem zlepšit tradiční metodami, jako je PSNR a MSE, které se ukázaly být v rozporu s vnímáním lidského oka. SSIM index je metoda pro měření podobnosti mezi dvěma obrazy, systém měření je na obr. 27. Tato metoda bere v potaz skutečnost, že lidské vidění je vysoce přizpůsobeno k extrahování strukturální informace. Index nabývá hodnot -1 až 1, kde 1 vyjadřuje shodné obrazy. U barevných obrazů se obvykle počítá jen na jasové složce  $Y$ . Hodnota SSIM je dána vztahem [13], [14]:

$$\text{SSIM}(x, y) = l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma, \quad (7)$$

kde

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \text{ je funkce porovnání jasu}, \quad (8)$$

$$c(x, y) = \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \text{ je funkce porovnání kontrastu}, \quad (9)$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \text{ je funkce porovnání struktury} \quad (10)$$

Při hodnotách  $\alpha = \beta = \gamma = 1$  a  $C_3 = C_2 / 2$  dostáváme konkrétní měřítko kvality SSIM:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (11)$$

kde

$\alpha, \beta, \gamma$  - kladné konstanty sloužící k váhování jednotlivých porovnávacích funkcí,

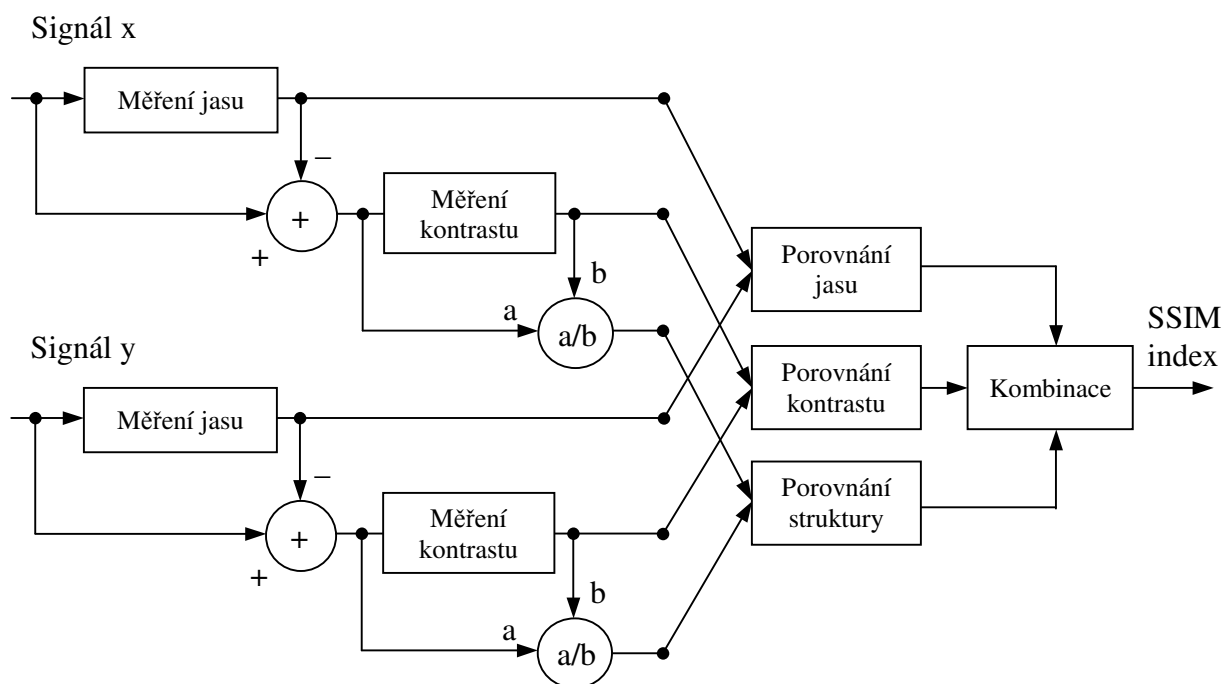
$x, y$  - souřadnice,

$\mu_x, \mu_y$  - vážený průměr  $x, y$ ,

$\sigma_x, \sigma_y$  - vážená variance  $x, y$ ,

$\sigma_{xy}$  - kovariance  $x$  a  $y$ ,

$C_i = (K_i L)^2$ ,  $L$  je maximální hodnota pixelu v obraze a  $K$  jsou malé konstanty. V práci jsou použité hodnoty  $K_1 = 0,01$  a  $K_2 = 0,03$ .



Obr. 27. Systém měření strukturální podobnosti [15].

## 6 Databáze komprimovaných videosekvencí

Databáze<sup>1</sup> byla vytvořena za pomoci programu FFmpeg 0.5 (komprimace MPEG-2 a AVC), VirtualDub 1.9.7 (build 32661/release), kodéru x264vfw – H.264/MPEG-4 AVC codec (komprimace AVC) a Windows Media Encoder 9 Series (komprimace VC-1). Databáze obsahuje komprimované videosekvence MPEG-2, AVC a VC-1 s nastavením různých profilů, bitové rychlosti (Bitrate), u AVC také různým nastavením rekonstrukčního filtru (In-loop deblocking filter). Testovací videosekvence jsou neprokládané, snímkování 50 snímků za sekundu, vzorkování 4:2:0, rozlišení 1280x720 a 1920x1080 [5].

Nastavované parametry: **Bitrate** – Hodnota průměrné bitové rychlosti. **Rekonstrukční filtr** – Obsahuje nastavení dvou hodnot: **Deblocking alpha** (Strength) a **Deblocking beta** (Threshold). První hodnota udává množství deblokování, které se na obraz použije, čím vyšší hodnota, tím více je obraz vyhlazený a ztrácí detaily. Druhá hodnota určuje jak citlivý je filtr na to, zda blok obsahuje detail nebo ne. U nižších hodnot se deblokování aplikuje méně na bloky s detaily a více na bloky bez detailů.

Nastavované profily: **MPEG-2** - Simple profile; High profile – standardní nastavení s přidaným parametrem pro B snímky. **AVC** - Baseline profile, Main profile a High profile. **VC-1** - Simple profile, Main profile a Advanced profile.

Byly komprimovány dvě videosekvence s rozdílnou náročností kódování - sekvence ParkJoy s obtížnou náročností kódování, viz obr. 28, a OldTownCross s jednoduchou náročností kódování, viz. obr 29.



Obr. 28. Ukázka snímku ze sekvence ParkJoy s obtížnou náročností kódování.

<sup>1</sup> Příložené DVD obsahuje databázi komprimovaných videosekvencí. Pro spuštění sekvencí je nutno mít přehrávač, např. VLC, a nainstalované příslušné kodeky, které jsou například součástí ffdshow.





Obr. 29. Ukázka snímku ze sekvence OldTownCross s jednoduchou náročností kódování.

Sekvence ParkJoy je video s pohybem ve scéně, obsahuje blízké předměty s detaily, odraz ve vodě. Posuv kamery kopíruje pohyb postav ve scéně. Naopak sekvence OldTownCross neobsahuje téměř žádný pohyb ve scéně, scéna obsahuje vzdálené objekty, téměř polovinu scény zabírá neměnná obloha. Posuv kamery je ve formě přeletu kamery nad městem.

Tabulka 5, 6 obsahuje výpisy komprimovaných videosekvencí, jejich bitové rychlosti a nastavený profil, pro formát 720p a 1080p. V tabulce 4 je výpis videosekvencí komprimovaných algoritmem AVC s různým nastavením rekonstrukčního filtru. Nastavení jednotlivých profilů pro algoritmy AVC a VC-1 je uvedeno v tabulce 7 a 8.

| Formát | Sekvence     | Deblocking alpha | Deblocking beta | Umístění<br>database\AVC\ |
|--------|--------------|------------------|-----------------|---------------------------|
| 720p   | ParkJoy      | bez filtru       |                 | InLoopFilter\720p\        |
|        |              | 0                | 0               |                           |
|        |              | 1                | 2               |                           |
|        | OldTownCross | bez filtru       |                 |                           |
|        |              | 0                | 0               |                           |
|        |              | -2               | -1              |                           |
| 1080p  | ParkJoy      | bez filtru       |                 | InLoopFilter\1080p\       |
|        |              | 0                | 0               |                           |
|        |              | 1                | 2               |                           |
|        | OldTownCross | bez filtru       |                 |                           |
|        |              | 0                | 0               |                           |
|        |              | -2               | -1              |                           |

Tab. 4. Nastavení rekonstrukčního filtru AVC.

| MPEG-2           |              |  |       |       |       |                             |
|------------------|--------------|--|-------|-------|-------|-----------------------------|
| Profil           | Sekvence     | Bitová rychlost [Mb/s]<br>sekvence č.: |       |       |       | Umístění<br>database\MPEG2\ |
|                  |              | 1                                      | 2     | 3     | 4     |                             |
| Simple profile   | ParkJoy      | 6,11                                   | 11,97 | 17,95 | 24,92 | SimpleProfile\720p\         |
|                  | OldTownCross | 6,06                                   | 12,03 | 18,02 | 25,01 |                             |
| High profile     | ParkJoy      | 6,42                                   | 11,89 | 17,90 | 24,88 | HighProfile\720p\           |
|                  | OldTownCross | 6,01                                   | 12,01 | 18,07 | 25,04 |                             |
|                  |              |  |       |       |       |                             |
| AVC              |              |  |       |       |       |                             |
| Profil           | Sekvence     | Bitová rychlost [Mb/s]<br>sekvence č.: |       |       |       | Umístění<br>database\AVC\   |
|                  |              | 1                                      | 2     | 3     | 4     |                             |
| Baseline profile | ParkJoy      | 4,00                                   | 9,99  | 15,00 | 25,01 | BaselineProfile\720p\       |
|                  | OldTownCross | 4,01                                   | 10,05 | 15,04 | 25,06 |                             |
| Main profile     | ParkJoy      | 4,01                                   | 10,09 | 15,09 | 25,02 | MainProfile\720p\           |
|                  | OldTownCross | 4,03                                   | 10,03 | 15,15 | 25,08 |                             |
| High profile     | ParkJoy      | 3,99                                   | 9,99  | 14,98 | 24,97 | HighProfile\720p\           |
|                  | OldTownCross | 4,03                                   | 10,10 | 15,14 | 25,09 |                             |
|                  |              |  |       |       |       |                             |
| VC-1             |              |  |       |       |       |                             |
| Profil           | Sekvence     | Bitová rychlost [Mb/s]<br>sekvence č.: |       |       |       | Umístění<br>database\VC1\   |
|                  |              | 1                                      | 2     | 3     | 4     |                             |
| Simple profile   | ParkJoy      | 2,31                                   | 6,60  | 13,42 | 18,83 | SimpleProfile\720p\         |
|                  | OldTownCross | 0,92                                   | 6,88  | 13,89 | 19,75 |                             |
| Main profile     | ParkJoy      | 2,28                                   | 6,52  | 13,67 | 18,23 | MainProfile\720p\           |
|                  | OldTownCross | 0,92                                   | 6,80  | 13,43 | 20,38 |                             |
| Advanced profile | ParkJoy      | 2,28                                   | 7,30  | 13,67 | 18,23 | AdvancedProfile\720p\       |
|                  | OldTownCross | 0,92                                   | 6,80  | 13,43 | 20,34 |                             |

Tab. 5. Výpis komprimovaných sekvencí pro formát 720p.



| MPEG-2           |              |  |       |       |       |                             |
|------------------|--------------|--|-------|-------|-------|-----------------------------|
| Profil           | Sekvence     | Bitová rychlost [Mb/s]<br>sekvence č.: |       |       |       | Umístění<br>database\MPEG2\ |
|                  |              | 1                                      | 2     | 3     | 4     |                             |
| Simple profile   | ParkJoy      | 11,00                                  | 11,97 | 17,96 | 24,99 | SimpleProfile\1080p\        |
|                  | OldTownCross | 6,06                                   | 12,04 | 18,02 | 25,01 |                             |
| High profile     | ParkJoy      | 12,49                                  | 12,49 | 17,92 | 24,95 | HighProfile\1080p\          |
|                  | OldTownCross | 5,98                                   | 12,08 | 18,13 | 25,09 |                             |
|                  |              |  |       |       |       |                             |
| AVC              |              |  |       |       |       |                             |
| Profil           | Sekvence     | Bitová rychlost [Mb/s]<br>sekvence č.: |       |       |       | Umístění<br>database\AVC\   |
|                  |              | 1                                      | 2     | 3     | 4     |                             |
| Baseline profile | ParkJoy      | 4,00                                   | 10,01 | 15,08 | 25,03 | BaselineProfile\1080p\      |
|                  | OldTownCross | 4,01                                   | 10,10 | 15,07 | 25,02 |                             |
| Main profile     | ParkJoy      | 4,00                                   | 10,03 | 15,02 | 25,02 | MainProfile\1080p\          |
|                  | OldTownCross | 4,02                                   | 10,01 | 15,21 | 25,09 |                             |
| High profile     | ParkJoy      | 3,99                                   | 9,99  | 15,00 | 25,00 | HighProfile\1080p\          |
|                  | OldTownCross | 4,03                                   | 10,03 | 15,06 | 25,11 |                             |
|                  |              |  |       |       |       |                             |
| VC-1             |              |  |       |       |       |                             |
| Profil           | Sekvence     | Bitová rychlost [Mb/s]<br>sekvence č.: |       |       |       | Umístění<br>database\VC1\   |
|                  |              | 1                                      | 2     | 3     | 4     |                             |
| Simple profile   | ParkJoy      | 4,86                                   | 8,92  | 14,29 | 17,95 | SimpleProfile\1080p\        |
|                  | OldTownCross | 0,93                                   | 7,18  | 13,52 | 18,87 |                             |
| Main profile     | ParkJoy      | 4,80                                   | 7,30  | 12,80 | 19,47 | MainProfile\1080p\          |
|                  | OldTownCross | 1,07                                   | 6,74  | 13,55 | 19,97 |                             |
| Advanced profile | ParkJoy      | 4,80                                   | 6,25  | 12,72 | 19,47 | AdvancedProfile\1080p\      |
|                  | OldTownCross | 1,07                                   | 6,74  | 13,56 | 19,97 |                             |

Tab. 6. Výpis komprimovaných sekvencí pro formát 1080p.

| Parametr                   |                                       | Profil              |                     |                     |
|----------------------------|---------------------------------------|---------------------|---------------------|---------------------|
|                            |                                       | Simple              | Main                | Advanced            |
|                            |                                       | Hodnota             |                     |                     |
| Dquant Option              | Diferenciální kvantizace              | -                   | -                   | vypnuto             |
| Dquant Method              | Metoda diferenciální kvantizace       | -                   | -                   | regulární           |
| In-Loop Filter             | Rekonstrukční filtr                   | -                   | zapnuto             | zapnuto             |
| Motion Search Level        | Úroveň odhadu pohybu                  | Fixed True Chroma   | Fixed True Chroma   | Fixed True Chroma   |
| Motion Search Range        | Vyhledávací oblast pro vektory pohybu | -                   | adaptivní           | adaptivní           |
| Motion Match Method        |                                       | adaptivní           | adaptivní           | adaptivní           |
| Macroblok Mode Cost        |                                       | Rate and Distortion | Rate and Distortion | Rate and Distortion |
| Delta Motion Vector Coding |                                       | -                   | -                   | Výchozí velikost    |
| Motion Vector Cost         |                                       | adaptivní           | adaptivní           | adaptivní           |
| B-Frame Number             | B snímky                              | -                   | 1                   | 1                   |
| B-Frame Delta QP           |                                       | -                   | 2                   | 2                   |
| Lookahead                  | Referenční snímky                     | 16                  | 16                  | 16                  |

Tab. 7. Nastavení profilů algoritmu VC-1.

| Parametr     |  | Profil                                       |  |  |
|--------------|--|--|--|--|
|              |  | Baseline                                     | Main   | High   |
|              |  | Hodnota                                      |  |  |
| coder        | CABAC kódování                                   |  | x  | x  |
| loop         | Rekonstrukční filtr                              | x  | x  | x  |
| partitions   | Povolené bloky                                   | parti8x8<br>parti4x4<br>partp8x8<br>partb8x8 | parti8x8<br>parti4x4<br>partp8x8<br>partb8x8 | parti8x8<br>parti4x4<br>partp8x8<br>partb8x8 |
| me_method    | Metoda odhadu pohybu                             | Hex  | Hex  | umh  |
| subq         | Výběr algoritmu pro subpixelovou predikci pohybu | 6  | 6  | 8  |
| me_range     | Vyhledávací oblast pro vektory pohybu            | 16   | 16   | 16   |
| g            | Velikost GOP                                     | 100  | 100  | 100  |
| keyint_min   | Minimální velikost GOP                           | 25   | 25   | 25   |
| sc_threshold | Práh pro detekci střihu                          | 40   | 40   | 40   |
| qcomp        | Parametr kompresní křivky                        | 0,6  | 0,6  | 0,6  |
| qmin         | Minimální hodnota kvantizačního parametru        | 10   | 10   | 10   |
| qmax         | Maximální hodnota kvantizačního parametru        | 51   | 51   | 51   |
| qdiff        | Maximální kvantizační krok                       | 4  | 4  | 4  |
| bf           | B snímky   |  | 4  | 4  |
| refs         | Referenční snímky                                |  | 2  | 4  |
| trellis      | Způsob zaokrouhlování kvantovaných koeficientů   |  | 0  | 1  |
| bpyramid     | Referenční B snímky                              |  | x  | x  |
| wpred        | Vážená předpověď B snímků                        |  | x  | x  |
| fastpskip    | Přeskočení makrobloků u P snímků                 | x  | x  | x  |
| dct8x8       | DCT 8x8  |  |  | x  |
| mixed_refs   | Smíšená reference                                |  |  | x  |

Tab. 8. Nastavení profilů algoritmu AVC.

**subq** – Tento parametr zapíná subpixelový odhad pohybu. Vyšší úrovně parametru zvýší kvalitu dalším zpracováním pohybové předpovědi, ale za cenu zvýšení času kódování. Standardní nastavení 6 zapíná funkci nazvanou optimalizace míry zkreslení, včetně psychovizuálních vylepšení. Nastavení 7 zapíná míru zkreslení pro B snímky. Nastavení 8 míru zkreslení zpřesňuje pro I a P snímky. Nastavení 9 přidává zpřesnění pro B [21].

## 7 Porovnání výkonnosti algoritmů

K porovnání výkonnosti, přesněji závislostem hodnoty PSNR a indexu SSIM na velikosti bitové rychlosti komprimovaného videosignálu byla použita sada testovacích videosekvencí ve vysokém rozlišení 1280x720 a 1920x1080 pixelů s neprokládaným řádkováním, snímkováním 50 snímků za sekundu a vzorkováním 4:2:0. Každá sada, jak u 720p, tak u 1080p obsahuje 5 videosekvencí s různým stupněm obtížnosti kódování. Sekvence CrownRun, ParkJoy, DucksTakeOff patří do kategorie s vysokou obtížností, zatímco sekvence InToTree a OldTownCross do kategorie s jednoduchou obtížností kódování [11].

K měření hodnot PSNR u algoritmů MPEG-2 a VC-1 byl použit Matlab R2009b a zdrojový kód [16] pro výpočet PSNR jasové složky a barevných složek. Použitá syntaxe příkazu:

```
function PSNR = yuvpsnr(File1,File2,width,height,format,parameter)
```

Pro hodnoty SSIM indexu byl pro tyto dva algoritmy použit VirtualubMod 1.5.10.1 spolu s AviSynth 2.5 a filtrem [17] v podobě dynamické knihovny pro výpočet SSIM, který byl vytvořen na základě výzkumu Zhou Wang [18]. Použitá syntaxe skriptu \*.avs:

```
A = Avisource("File1.avi")
B = Avisource("File2.avi")

return ssim(A,B,"vysledky.csv","avg_SSIM.txt",lumimask=false)
```

U algoritmu AVC byl použit program FFmpeg 0.5 a to jak pro PSNR, tak pro index SSIM. Výpočet se provedl při samotné komprimaci za pomoci parametru “-psnr”. Použitá syntaxe příkazu:

```
ffmpeg.exe -i File1.yuv -f rawvideo -g 100 -pass 1 -vcodec libx264
-vpre hq -b 3145728 -bt 100000 -threads 0 -psnr File2.avi
```

### 7.1 Testovací videosekvence formátu 720p

Videosekvence s rozlišením 1280x720 délky 10 sekund byly u MPEG-2 a AVC komprimovány s nastavením High Profile, u VC-1 pak Advanced Profile. Ke komprimaci algoritmy AVC a MPEG-2 byl použit program FFmpeg 0.5, u AVC je rozsah 1-25Mb/s, u MPEG-2 pak 6-25Mb/s. Pro komprimaci algoritmem VC-1 byl použit program Windows Media Encoder a rozsah komprimací je od jednotek Mb/s do 20Mb/s.

#### 7.1.1 Závislosti PSNR jasové složky Y na bitové rychlosti

Následujících 5 grafů, viz obr. 30, 31, 32, 33, 34, zobrazuje srovnání algoritmů pro jednotlivé videosekvence. V tabulce 9 jsou pro příklad uvedeny hodnoty bitových rychlostí jednotlivých algoritmů pro komprimované videosekvence se stejnou kvalitou.

U sekvence CrowdRun dosahoval nejlepších hodnot algoritmus AVC. Algoritmus VC-1 byl na nižších bitových rychlostech horší o několik desetin decibelu, na vyšších

hodnotách bitové rychlosti až o 1dB. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, hodnoty ovšem byly na celém společném rozsahu menší o 2,5dB až 3dB.

U sekvence ParkJoy byl průběh podobný jako u sekvence CrowdRun, s rozdílem, že algoritmus VC-1 byl horší než algoritmus AVC o téměř konstantní hodnotu (0,8dB – 1dB) na celém společném rozsahu. Rozdíl mezi algoritmem MPEG-2 a AVC se zmenšil na 2dB až 2,5dB.

U sekvence DucksTakeOff měly algoritmy AVC a VC-1 stejný průběh, až na vyšších bitových rychlostech, zhruba 15Mb/s - 20Mb/s, klesly hodnoty VC-1 pod AVC o cca 0,5dB. Rozdíl mezi algoritmem MPEG-2 a AVC byl konstantní na celém společném rozsahu a to o cca 1,8dB.

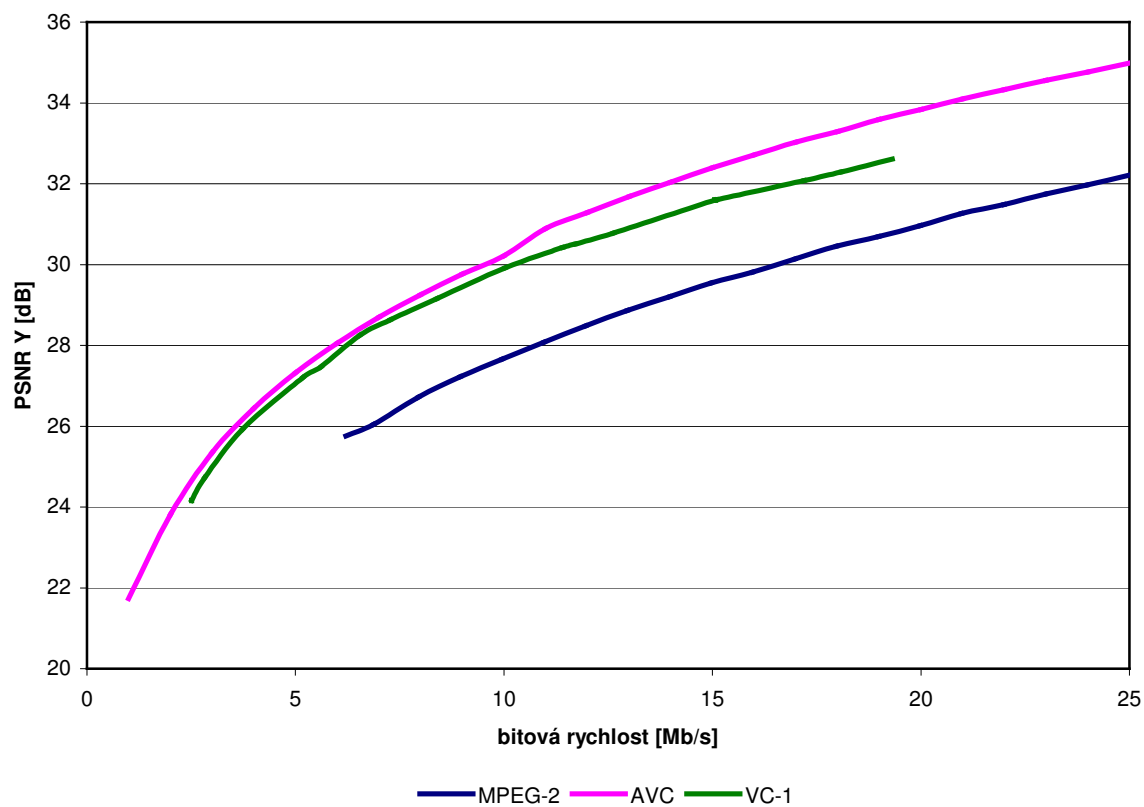
U sekvence InToTree měly algoritmy AVC a VC-1 stejný průběh na rozsahu 1Mb/s až 5Mb/s, s rostoucí bitovou rychlostí pak začaly hodnoty VC-1 lineárně klesat až na rozdíl cca 0,7dB. Rozdíl mezi algoritmem MPEG-2 a AVC byl konstantní na celém společném rozsahu a to o cca 1,1dB.

U sekvence OldTownCross dosahoval na rozsahu 1Mb/s až 5Mb/s algoritmus VC-1 hodnoty o cca 0,8dB menší než algoritmus AVC, naopak na rozsahu 5Mb/s až 17Mb/s dosahoval algoritmus VC-1 hodnoty o cca 0,4dB vyšší než algoritmus AVC. Rozdíl mezi algoritmem MPEG-2 a AVC byl na nižších bitových rychlostech až 2,8dB, s rostoucí bitovou rychlostí se zmenšoval, až na desetiny decibelu při hodnotě 20Mb/s.

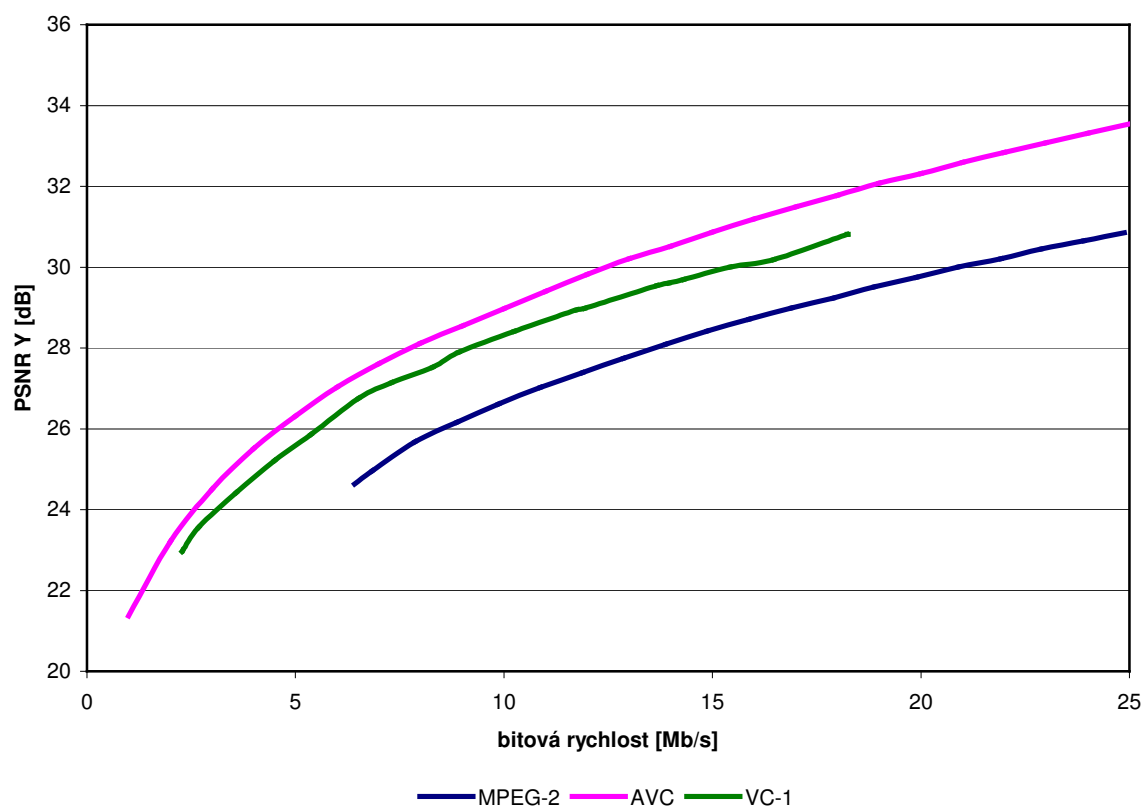
**Shrnutí** – Pro sekvence s vyšší obtížností kódování dosahuje nejlepších hodnot algoritmus AVC. Algoritmus VC-1 se při nízkých bitových rychlostech vyrovnává nebo přibližuje v kvalitě algoritmu AVC, při vyšších rychlostech však kvalita oproti AVC klesá. Algoritmus MPEG-2 dosahuje konstantně nejhorších výsledků. U sekvencí s nižší obtížností kódování jsou rozdíly mezi algoritmy AVC a VC-1 malé, u vyšších bitových rychlostí dosahují dokonce všechny tři algoritmy přibližně stejné kvality.

| Sekvence     | Obtížnost kódování | PSNR Y [dB] | Bitová rychlost [Mb/s] |      |        |
|--------------|--------------------|-------------|------------------------|------|--------|
|              |                    |             | AVC                    | VC-1 | MPEG-2 |
| CrowdRun     | obtížná            | 30          | 9,5                    | 10,5 | 16,5   |
| DucksTakeOff | obtížná            | 30          | 13                     | 14   | 19     |
| ParkJoy      | obtížná            | 30          | 12,5                   | 15,5 | 21     |
| InToTree     | snadná             | 36          | 9,5                    | 11   | 14,5   |
| OldTownCross | snadná             | 36          | 3                      | 4    | 9,5    |

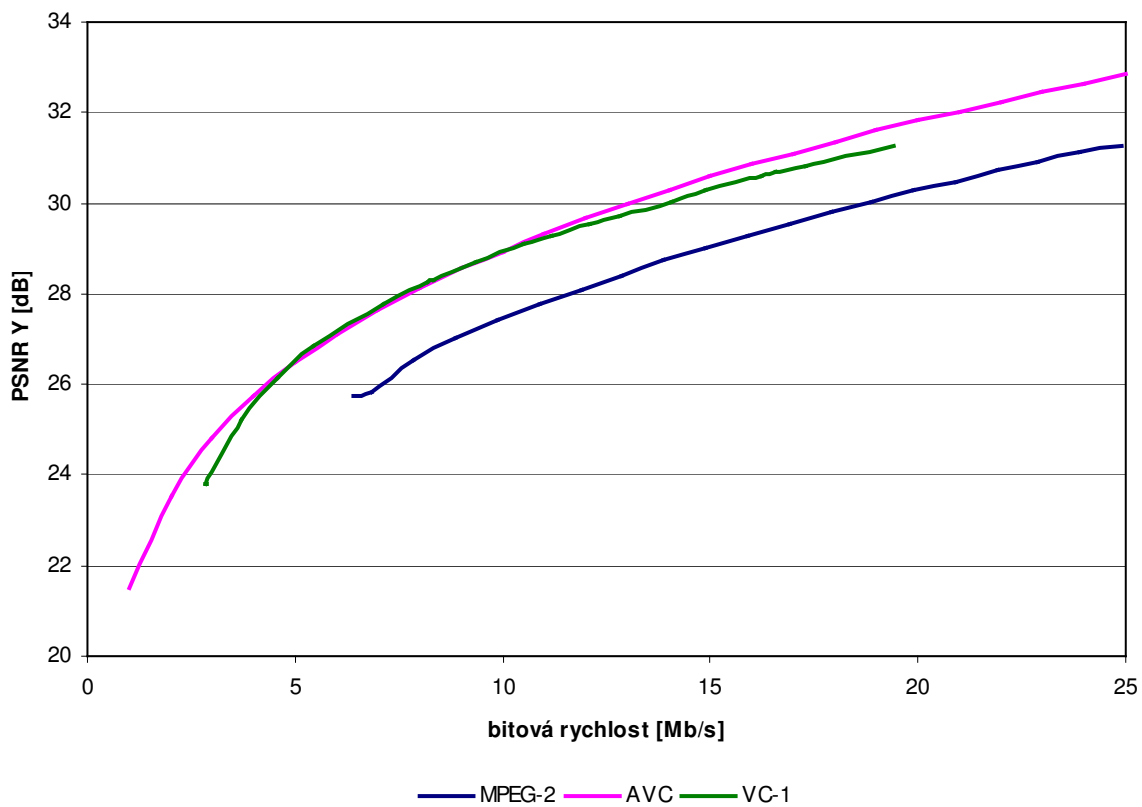
Tab. 9. Srovnání bitové rychlosti algoritmů při shodné kvalitě (PSNR/720p).



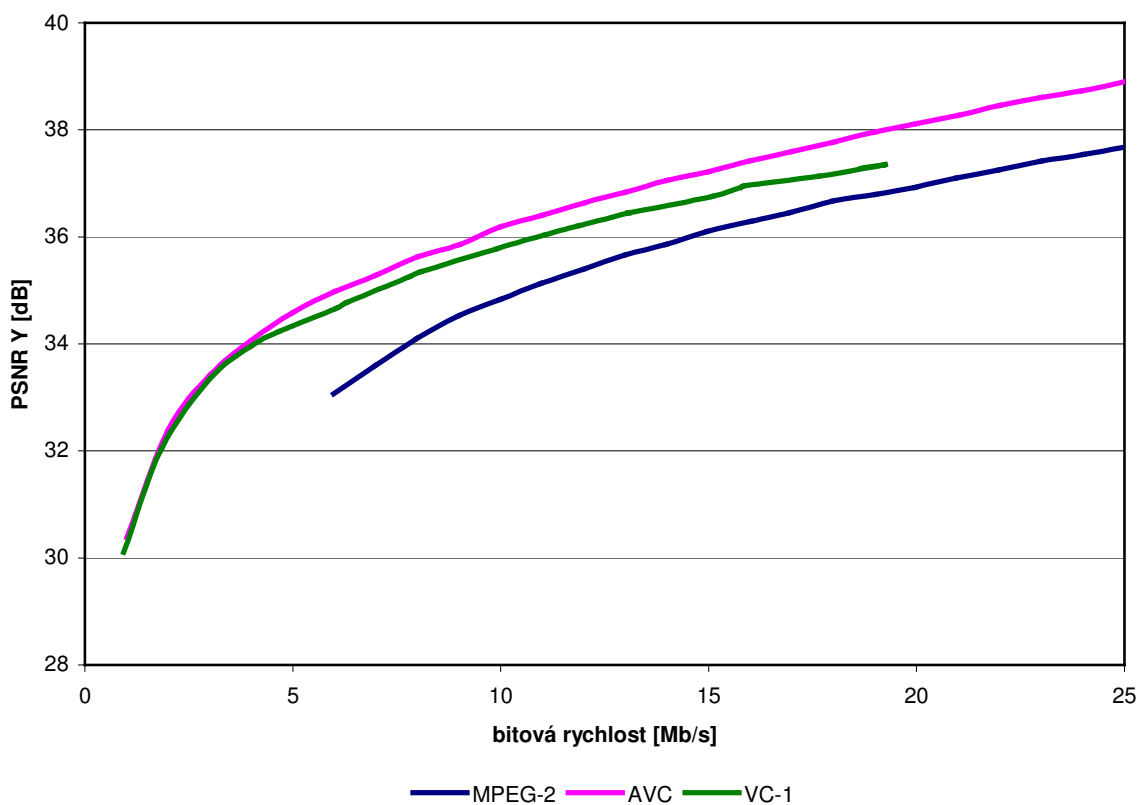
Obr. 30. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci CrowdRun.



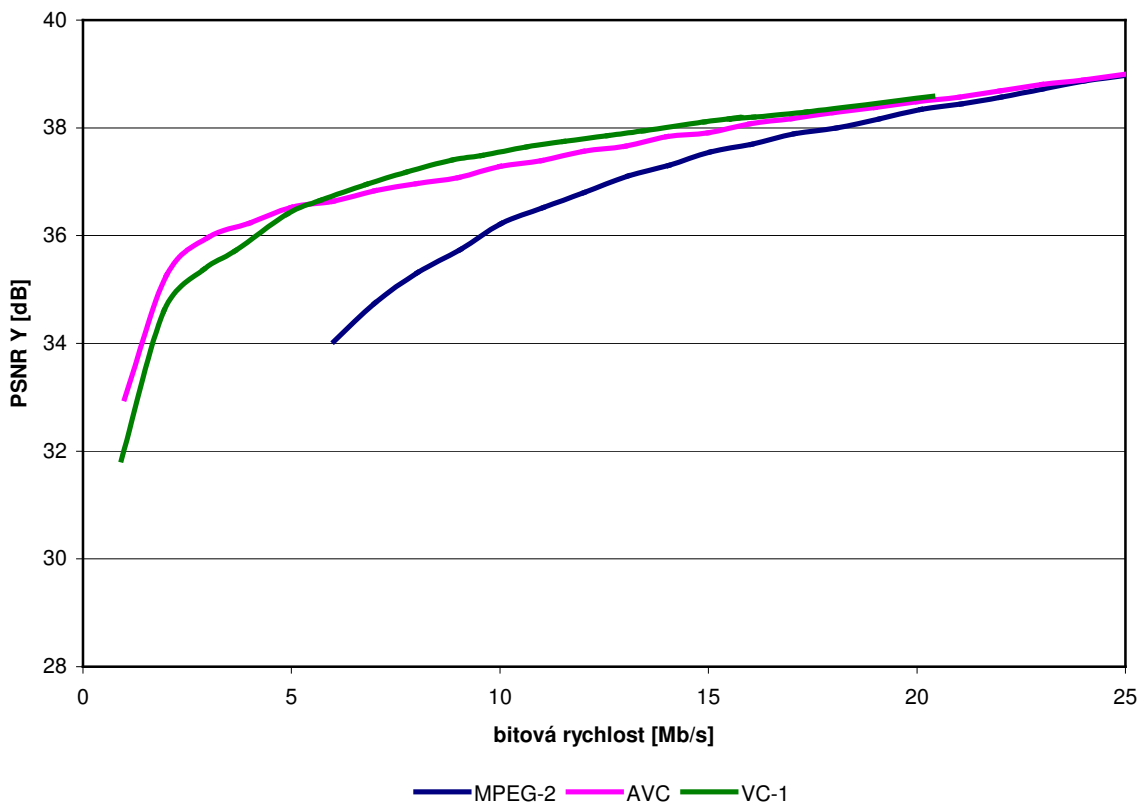
Obr. 31. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci ParkJoy.



Obr. 32. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci DucksTakeOff.



Obr. 33. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci InToTree.



Obr. 34. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci OldTownCross.

### 7.1.2 Závislosti indexu SSIM na bitové rychlosti

Následujících 5 grafů, viz obr. 35, 36, 37, 38, 39, zobrazuje srovnání algoritmů pro jednotlivé videosekvence. V tabulce 10 jsou pro příklad uvedeny hodnoty bitových rychlostí jednotlivých algoritmů pro komprimované videosekvence se stejnou hodnotou indexu SSIM.

U sekvence CrowdRun dosahoval nejlepších hodnot algoritmus AVC. Algoritmus VC-1 měl na celém společném rozsahu s AVC hodnoty indexu SSIM menší o cca 0,03. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, ale dosahoval menších hodnot než AVC. Rozdíl na nižších bitových rychlostech měl hodnotu až 0,1 a s rostoucí bitovou rychlost se zmenšoval, až na cca 0,05 při nejvyšší hodnotě bitové rychlosti – 25Mb/s.

U sekvence ParkJoy byl průběh podobný jako u sekvence CrowdRun. Algoritmus VC-1 měl na celém společném rozsahu s AVC hodnoty indexu SSIM menší o cca 0,05. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, ale dosahoval menších hodnot než AVC. Rozdíl na nižších bitových rychlostech měl hodnotu 0,13 a s rostoucí bitovou rychlost se zmenšoval, až na cca 0,05 při nejvyšší hodnotě bitové rychlosti – 25Mb/s.

U sekvence DucksTakeOff dosahoval algoritmus VC-1 na nižších hodnotách bitové rychlosti hodnoty indexu SSIM menší o cca 0,05 než AVC, na vyšších hodnotách bitové rychlosti pak rozdíl klesal až na cca 0,03. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, ale dosahoval menších hodnot než AVC. Rozdíl u nižších bitových rychlostí měl hodnotu až 0,08 a s rostoucí bitovou rychlostí se zmenšoval, až na cca 0,04 při nejvyšší hodnotě bitové rychlosti – 25Mb/s.

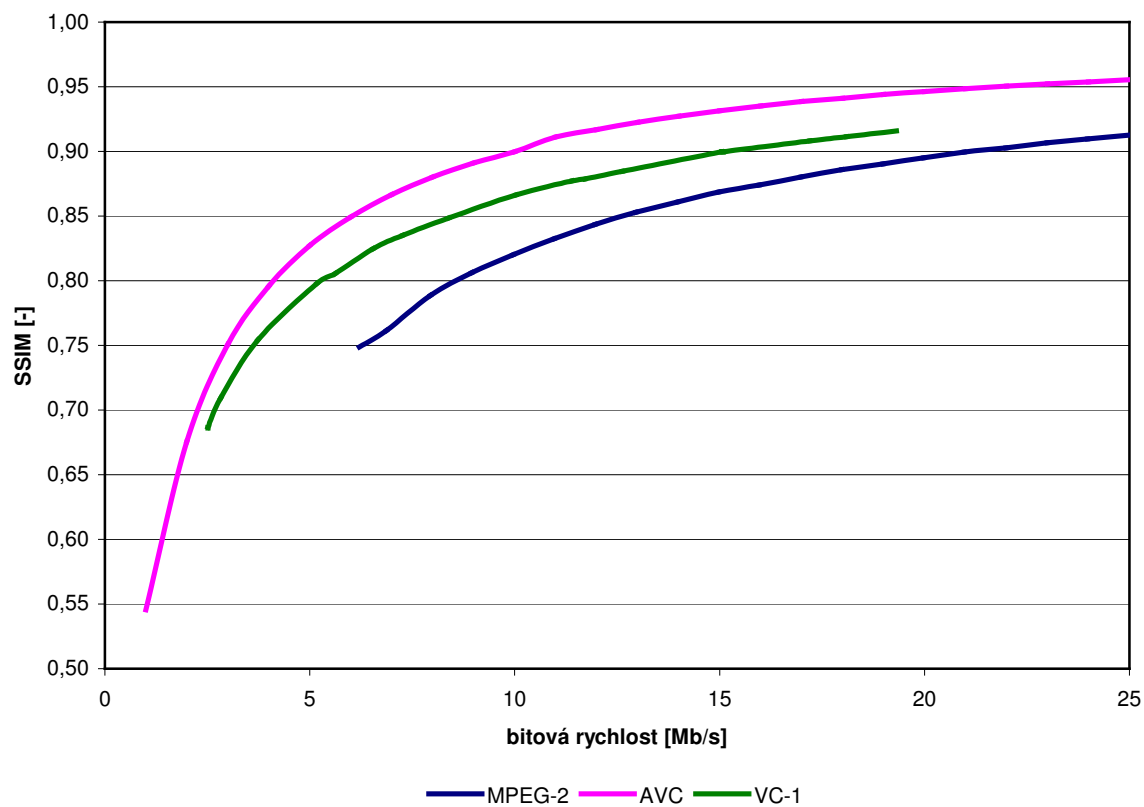


U sekvence InToTree dosahoval algoritmus VC-1 na nižších hodnotách bitové rychlosti téměř stejný průběh jako u algoritmu AVC. Od 5Mb/s do 20Mb/s měl algoritmus VC-1 hodnoty menší než AVC, o hodnotu cca 0,02. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, ale dosahoval menších hodnot než AVC. Rozdíl na nižších bitových rychlostech dosahoval hodnoty až 0,1 a s rostoucí bitovou rychlostí se zmenšoval, až na cca 0,02 při nejvyšší hodnotě bitové rychlosti – 25Mb/s.

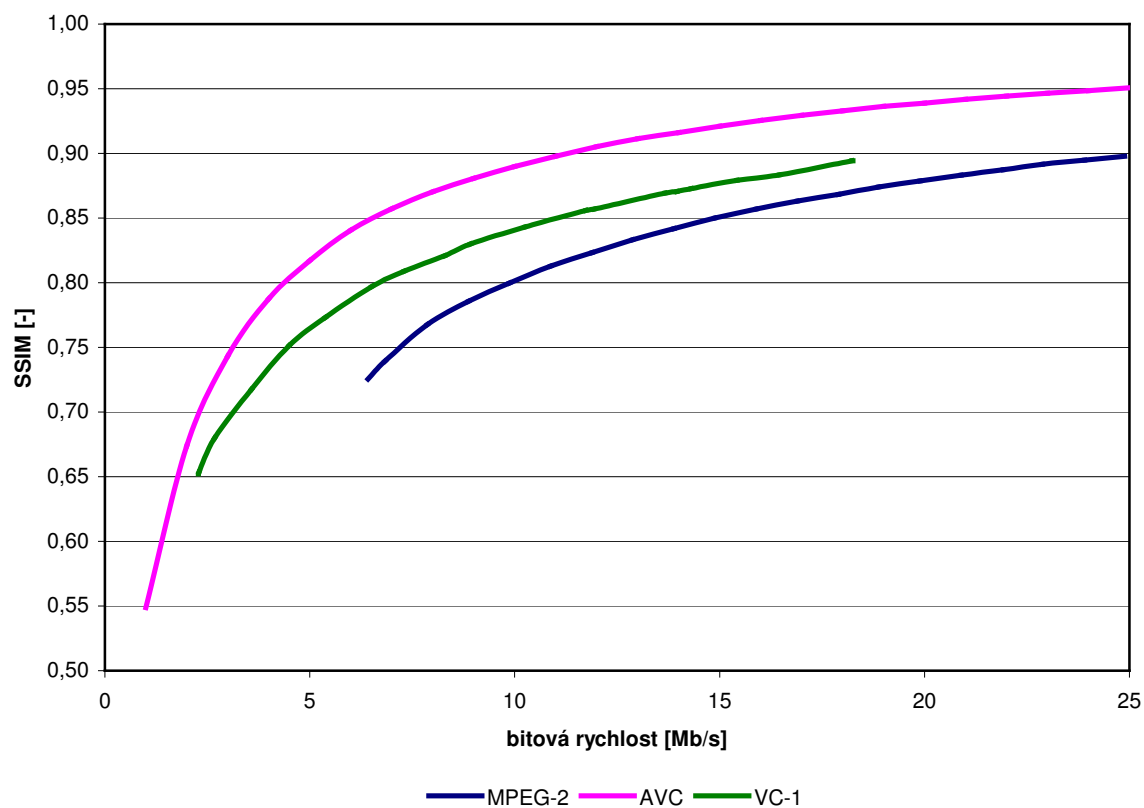
U sekvence OldTownCross měly algoritmy AVC a VC-1 podobný průběh, na rozsahu 1Mb/s až 4Mb/s měl algoritmus VC-1 hodnoty o cca 0,02 menší než algoritmus AVC, od 4Mb/s vyše byly průběhy téměř totožné. Rozdíl mezi algoritmem MPEG-2 a AVC byl na rozsahu 6Mb/s až 12Mb/s 0,03. Od 12Mb/s měly všechny tři algoritmy téměř stejný průběh.

| Sekvence     | Obtížnost kódování | SSIM [-] | Bitová rychlost [Mb/s] |      |        |
|--------------|--------------------|----------|------------------------|------|--------|
|              |                    |          | AVC                    | VC-1 | MPEG-2 |
| CrowdRun     | obtížná            | 0,9      | 10                     | 15   | 21     |
| DucksTakeOff | obtížná            | 0,9      | 11,5                   | 16   | 20     |
| ParkJoy      | obtížná            | 0,9      | 11,5                   | 19   | 25     |
| InToTree     | snadná             | 0,9      | 6,5                    | 9    | 12     |
| OldTownCross | snadná             | 0,9      | 1,2                    | 1,6  | 5      |

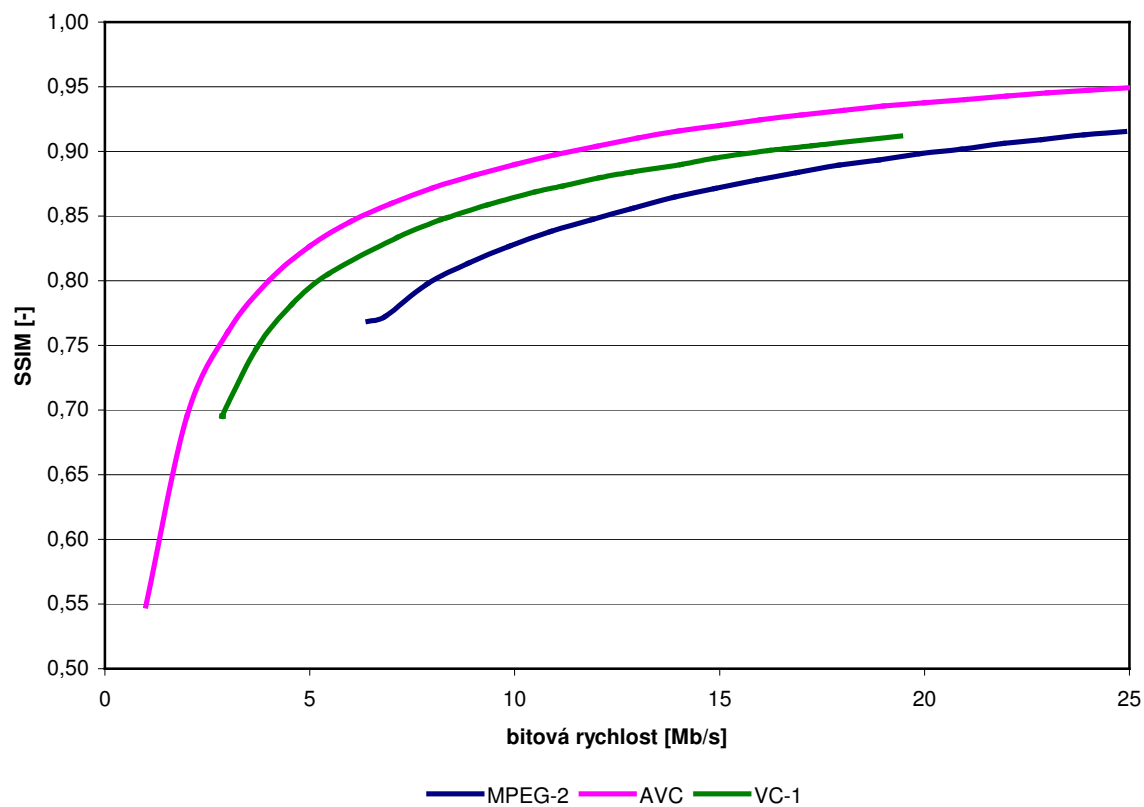
Tab. 10. Srovnání bitové rychlosti algoritmů při shodné kvalitě (SSIM/720p).



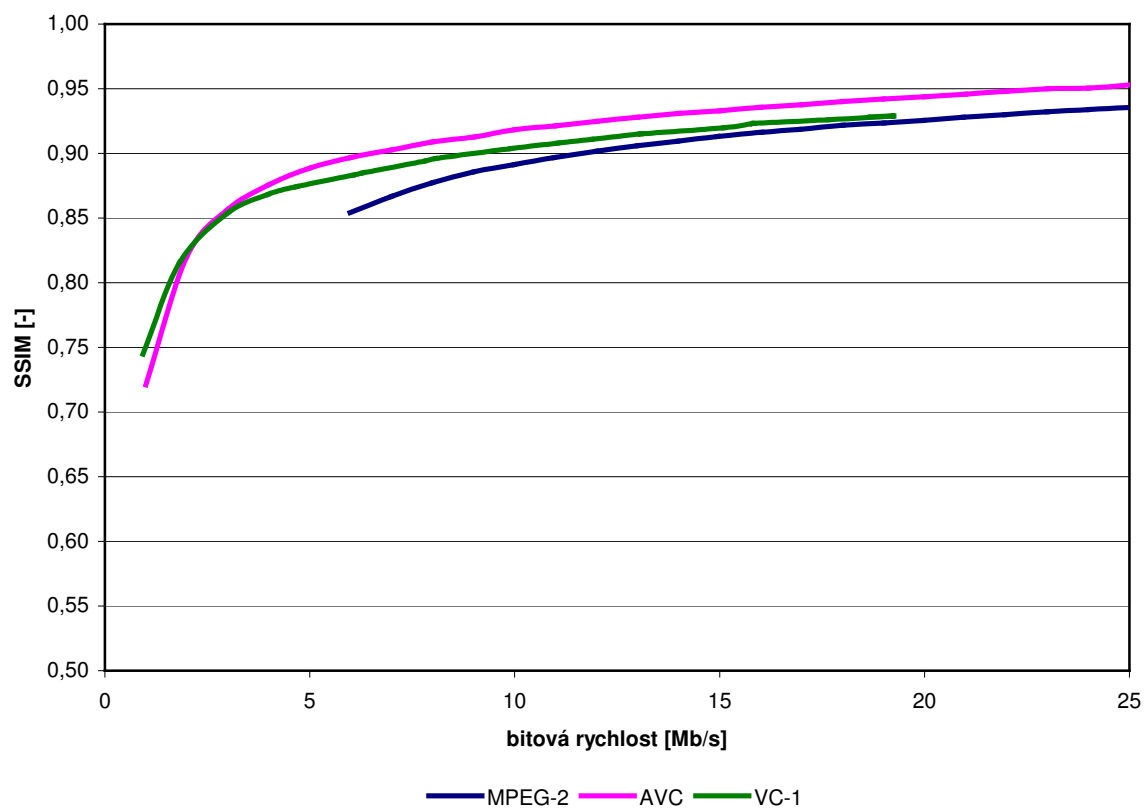
Obr. 35. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci CrowdRun.



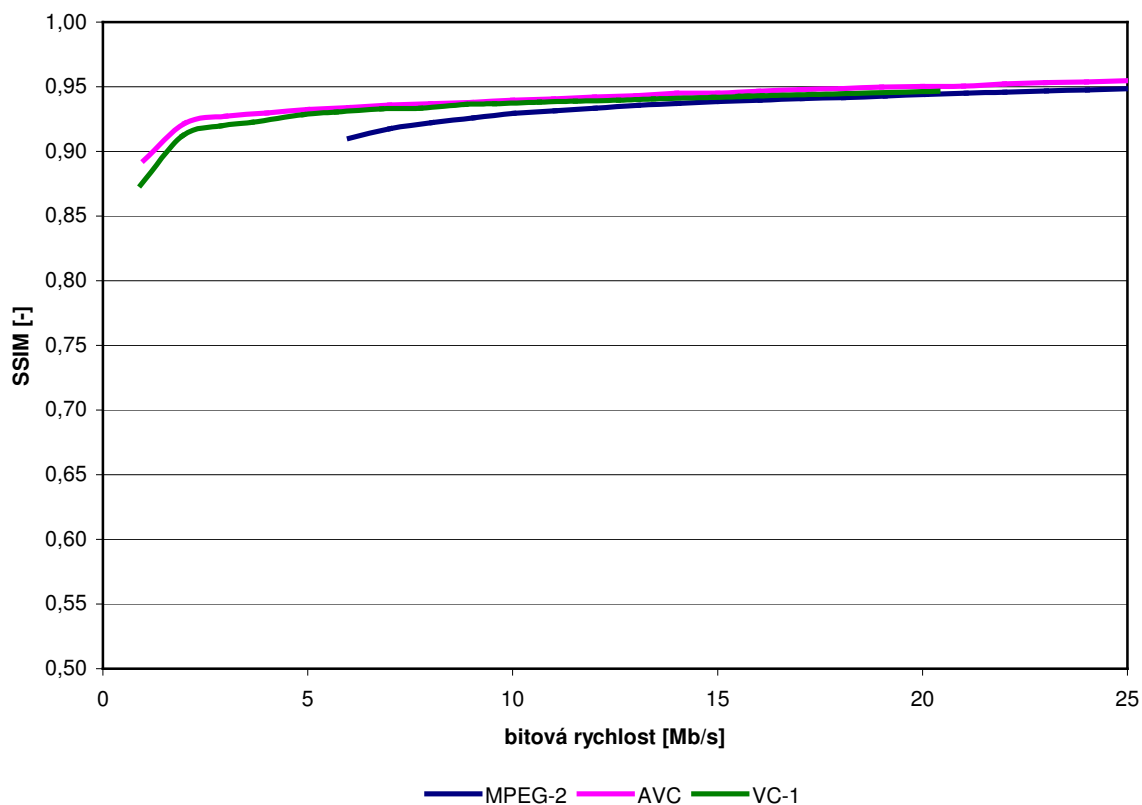
Obr. 36. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci ParkJoy.



Obr. 37. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci DucksTakeOff.



Obr. 38. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci InToTree.



Obr. 39. Srovnání závislosti indexu SSIM na bitové rychlosti pro sekvenci OldTownCross.

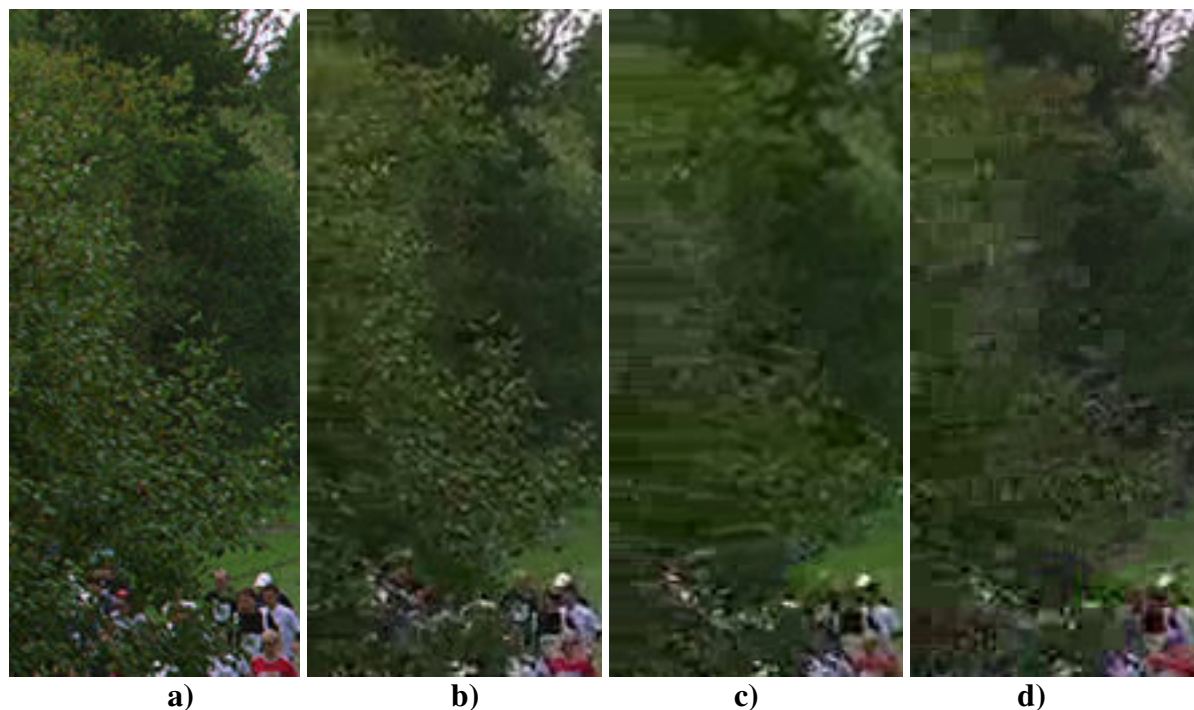
### 7.1.3 Porovnání subjektivní kvality

Jedná se o srovnání originálního, nekomprimovaného snímku se snímky komprimovanými jednotlivými algoritmy, viz obr. 40. Pro porovnání subjektivní kvality byl vybrán snímek 75 ze sekvence ParkJoy, komprimované sekvence dosahovaly přibližně stejné hodnoty PSNR = 28 dB.



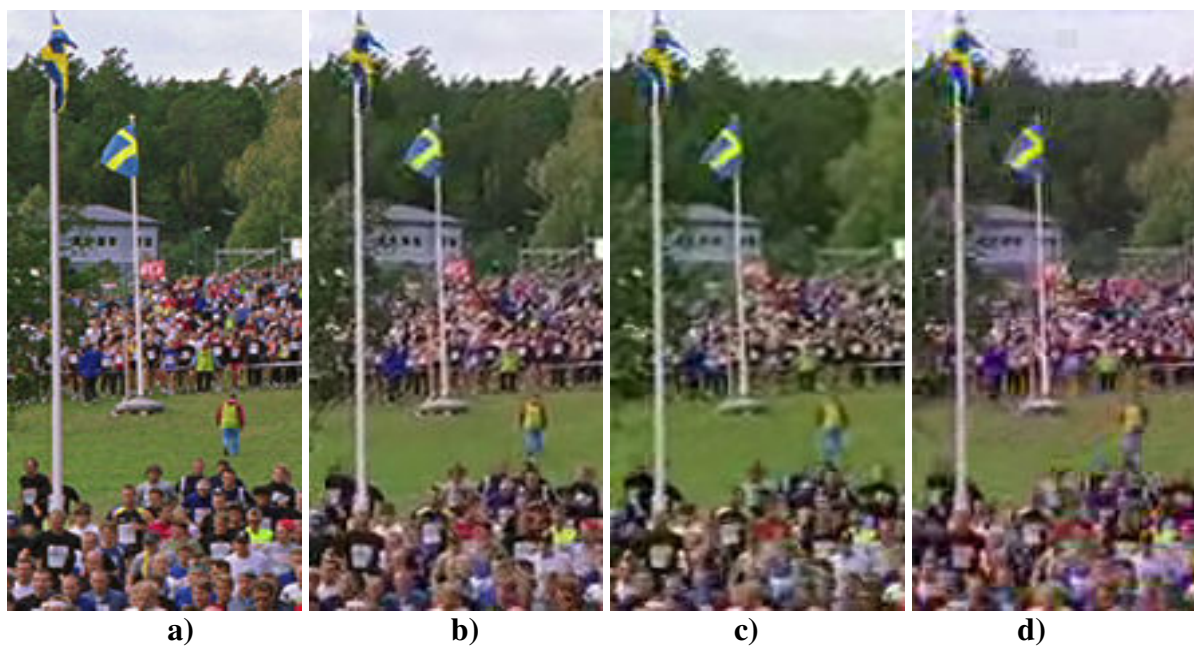
Obr. 40. Originální snímek s vybranými oblastmi pro porovnání – 720p.

Oblast 1, viz obr. 41 – Algoritmus AVC vykazuje částečné rozmazání obrazu na levém okraji oblasti. U algoritmu VC-1 je rozmazání obrazu velmi výrazné, objevují se blokové artefakty. U algoritmu MPEG-2 se vytváří blokové artefakty.



Obr. 41. Porovnání kvality snímků – oblast 1 **a)** originální **b)** AVC **c)** VC-1 **d)** MPEG-2.

Oblast 2 viz obr. 42 – Algoritmus AVC vykazuje vyhlazené bloky, čím se částečné ztrácí detaily. U algoritmu VC-1 se objevují blokové artefakty, které jsou z velké části vyhlazeny. U algoritmu MPEG-2 se vytváří blokové artefakty, dochází ke ztrátě detailů a nepřesnostem v obraze.

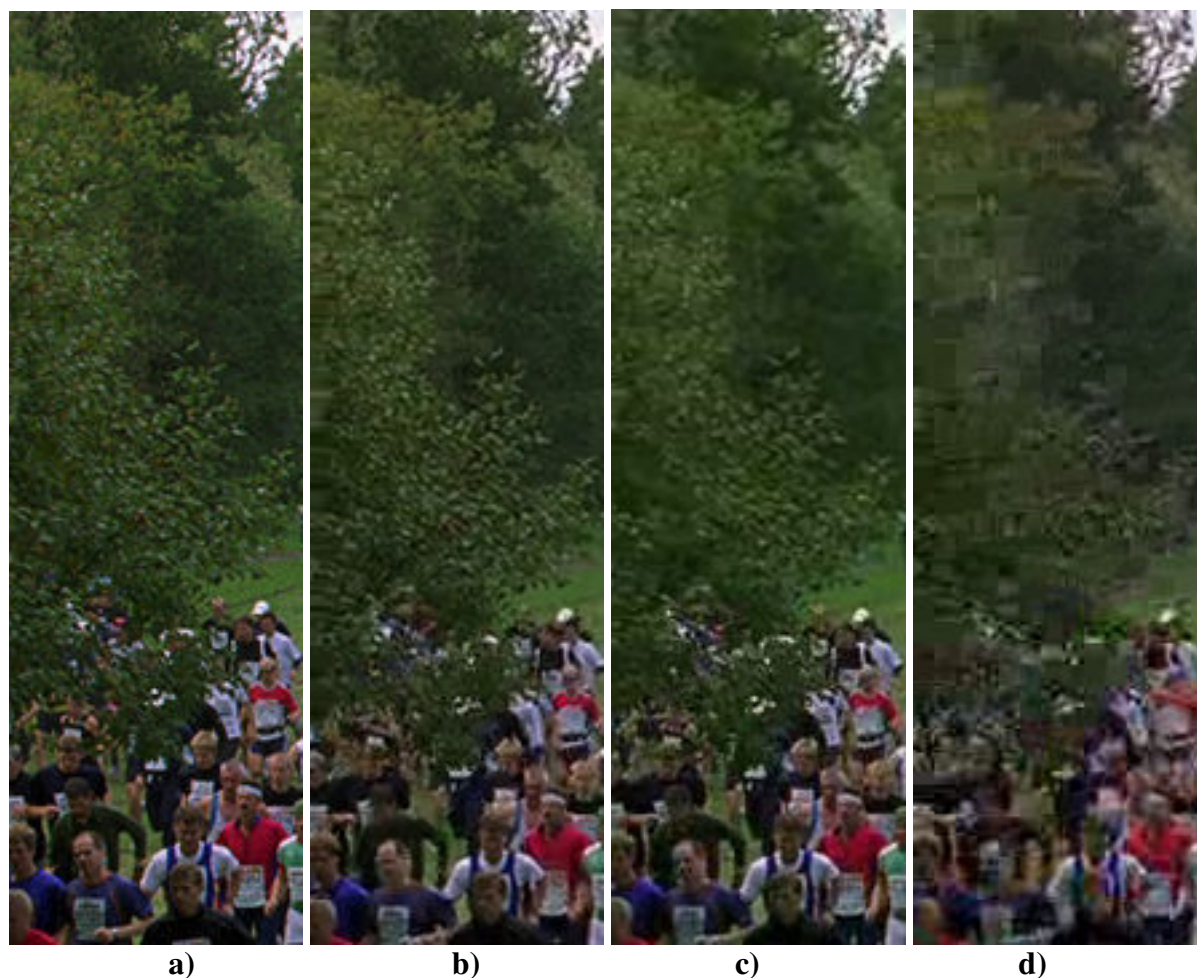


Obr. 42. Porovnání kvality snímků – oblast 2 **a)** originální **b)** AVC **c)** VC-1 **d)** MPEG-2.



Pro srovnání byly vybrány i snímky ze sekvencí o přibližně stejné bitové rychlosti 10Mb/s, ke srovnání kvality byl použit stejný snímek (75) ze sekvence ParkJoy.

Oblast 1, viz obr. 43 – Algoritmus AVC vykazuje velmi malé rozmazání obrazu na levém okraji oblasti. Algoritmus VC-1 vykazuje oproti AVC o něco větší vyhlazení obrazu. U algoritmu MPEG-2 se vytváří blokové artefakty.



Obr. 43. Porovnání kvality snímků – oblast 1 **a)** originální **b)** AVC **c)** VC-1 **d)** MPEG-2.

## 7.2 Testovací videosekvence formátu 1080p

Videosekvence s rozlišením 1920x1080 délky 10 sekund, až na sekvenci InToTree o délce 3 sekundy, byly u MPEG-2 a AVC komprimovány s nastavením High Profile, u VC-1 pak Advanced Profile. Ke komprimaci algoritmy AVC a MPEG-2 byl použit program Ffmpeg 0.5, u AVC je rozsah 2-25Mb/s, u MPEG-2 se pohybuje v závislosti na sekvenci od 7-14Mb/s do 25Mb/s. Pro komprimaci algoritmem VC-1 byl použit program Windows Media Encoder a rozsah komprimací je od jednotek Mb/s do 20Mb/s.

### 7.2.1 Závislosti PSNR jasové složky Y na bitové rychlosti

Následujících 5 grafů, viz obr. 44, 45, 46, 47, 48, zobrazuje srovnání algoritmů pro jednotlivé videosekvence. V tabulce 11 jsou pro příklad uvedeny hodnoty bitových rychlostí jednotlivých algoritmů pro komprimované videosekvence se stejnou kvalitou.

U sekvence CrowdRun byly průběhy algoritmů AVC a VC-1 totožné. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, hodnoty byly ovšem na celém společném rozsahu menší o 2,5dB.

U sekvence ParkJoy dosahoval nejlepších hodnot algoritmus AVC, algoritmus VC-1 měl na celém společném rozsahu s AVC hodnoty menší o cca 0,5dB. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, hodnoty ovšem byly na celém společném rozsahu menší o 2dB.

U sekvence DucksTakeOff měl v rozsahu 6Mb/s až 8Mb/s lepší hodnoty algoritmus AVC. Od 8Mb/s a výše dosahoval lepších hodnot algoritmus VC-1, o konstantní hodnotu 0,2dB. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC, hodnoty ovšem byly na celém společném rozsahu menší o cca 1,5dB.

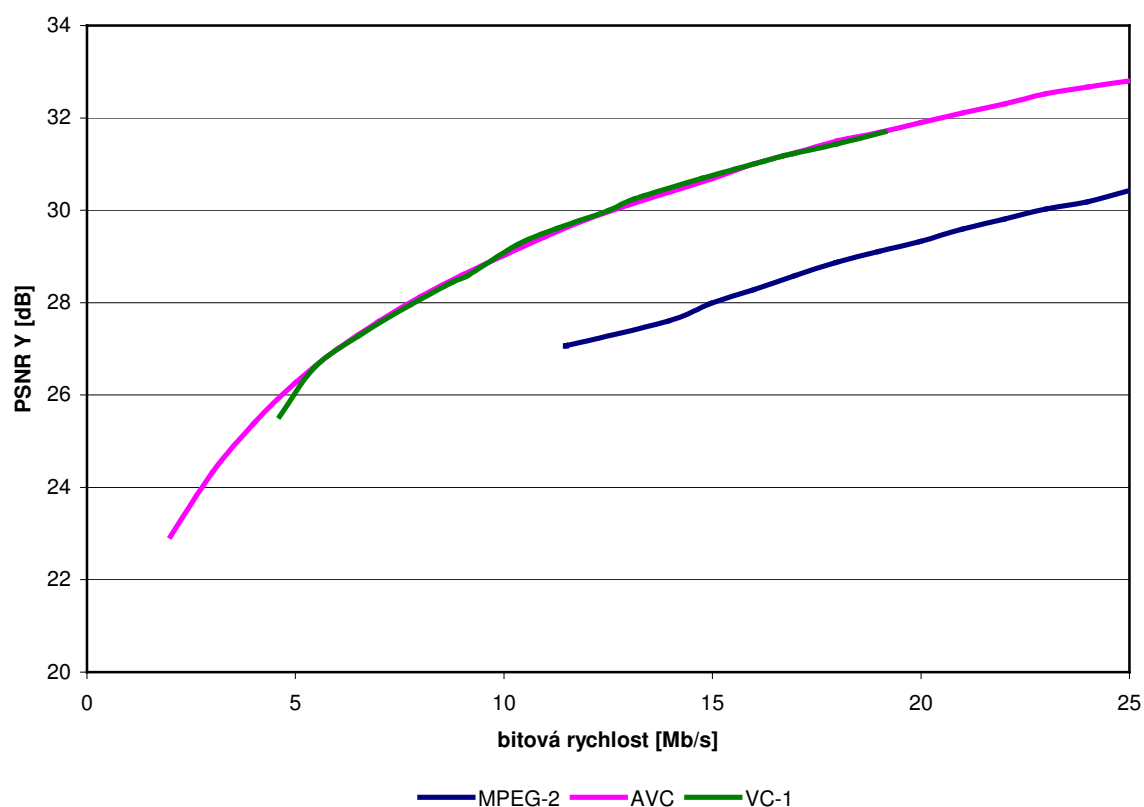
U sekvence InToTree dosahoval nejlepších hodnot algoritmus VC-1. Algoritmus AVC dosahoval hodnot menších o 0,8dB na celém společném rozsahu. Algoritmus MPEG-2 dosahoval na nižších bitových rychlostech hodnot menších než algoritmus AVC, s rostoucí bitovou rychlostí rozdíl klesal, až do cca 18Mb/s, kdy začaly být průběhy algoritmů MPEG-2 a AVC totožné.

U sekvence OldTownCross byly výsledky totožné se sekvencí InToTree.

**Shrnutí** – Algoritmus VC-1 dosahuje, na rozdíl od formátu 720p, u všech videosekvencí, až na jednu výjimku, stejných nebo vyšších hodnot než algoritmus AVC. Stejně jako u formátu 720p dosahuje i v tomto případě algoritmus MPEG-2 nejhorší kvality u videosekvencí s vyšší obtížností kódování, u videosekvencí s nižší obtížností kódování dosahují u vyšších bitových rychlostí všechny tři algoritmy přibližně stejné kvality, obdobně jako u formátu 720p.

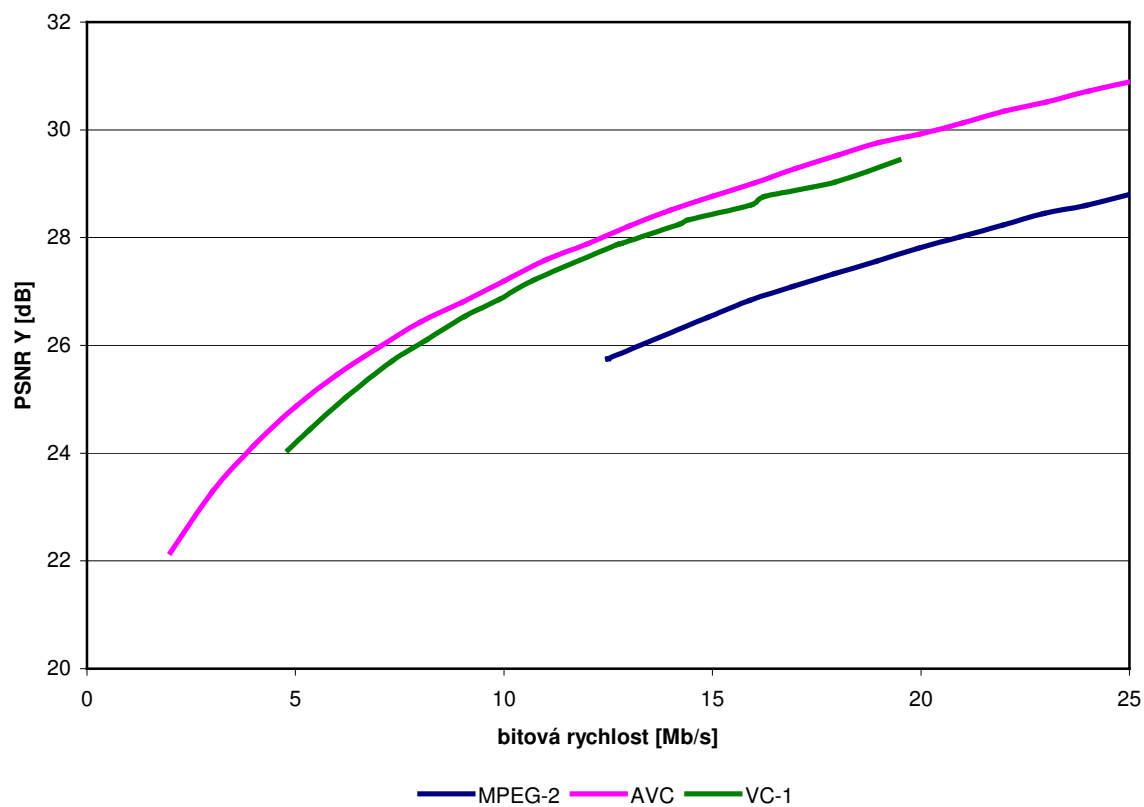
| Sekvence     | Obtížnost kódování | PSNR Y [dB] | Bitová rychlost [Mb/s] |      |        |
|--------------|--------------------|-------------|------------------------|------|--------|
|              |                    |             | AVC                    | VC-1 | MPEG-2 |
| CrowdRun     | obtížná            | 28          | 8                      | 8    | 15     |
| DucksTakeOff | obtížná            | 28          | 12,2                   | 13   | 20,4   |
| ParkJoy      | obtížná            | 28          | 12,3                   | 13,2 | 21     |
| InToTree     | snadná             | 34          | 6,5                    | 3    | 13,5   |
| OldTownCross | snadná             | 34          | 4                      | 3,5  | 10,5   |

Tab. 11. Srovnání bitové rychlosti algoritmů při shodné kvalitě (PSNR/1080p).

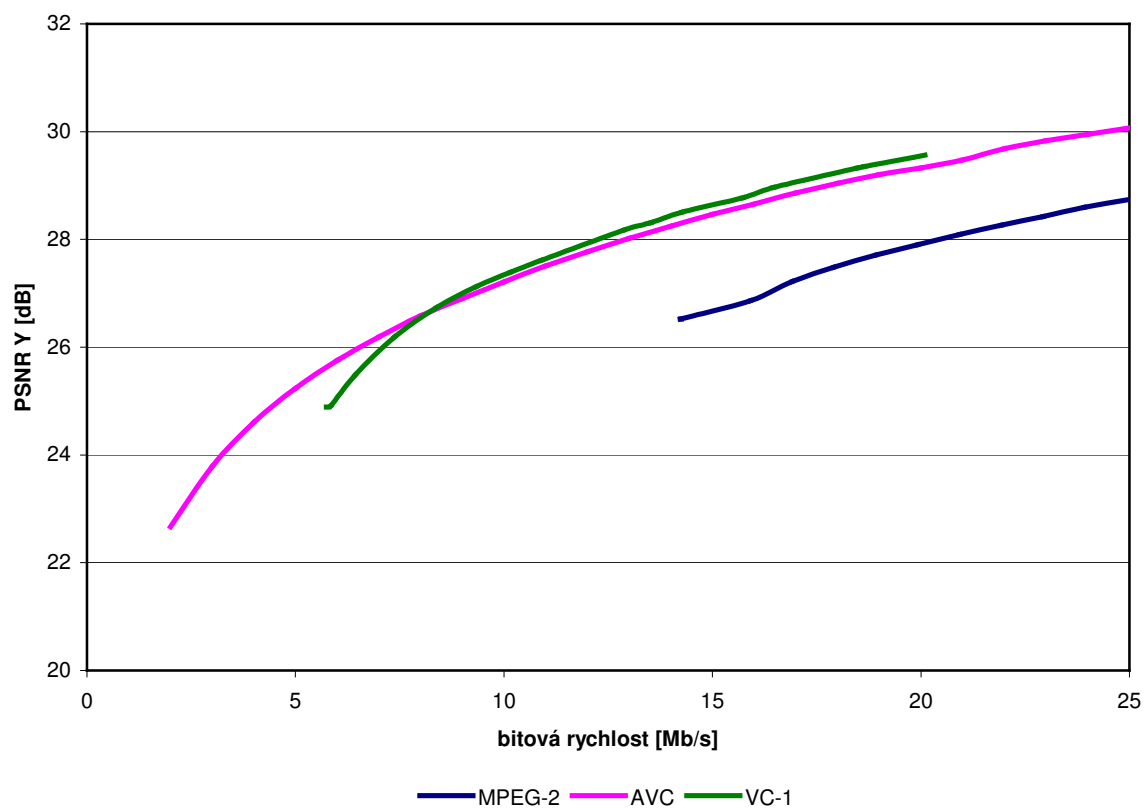


Obr. 44. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci CrowdRun.

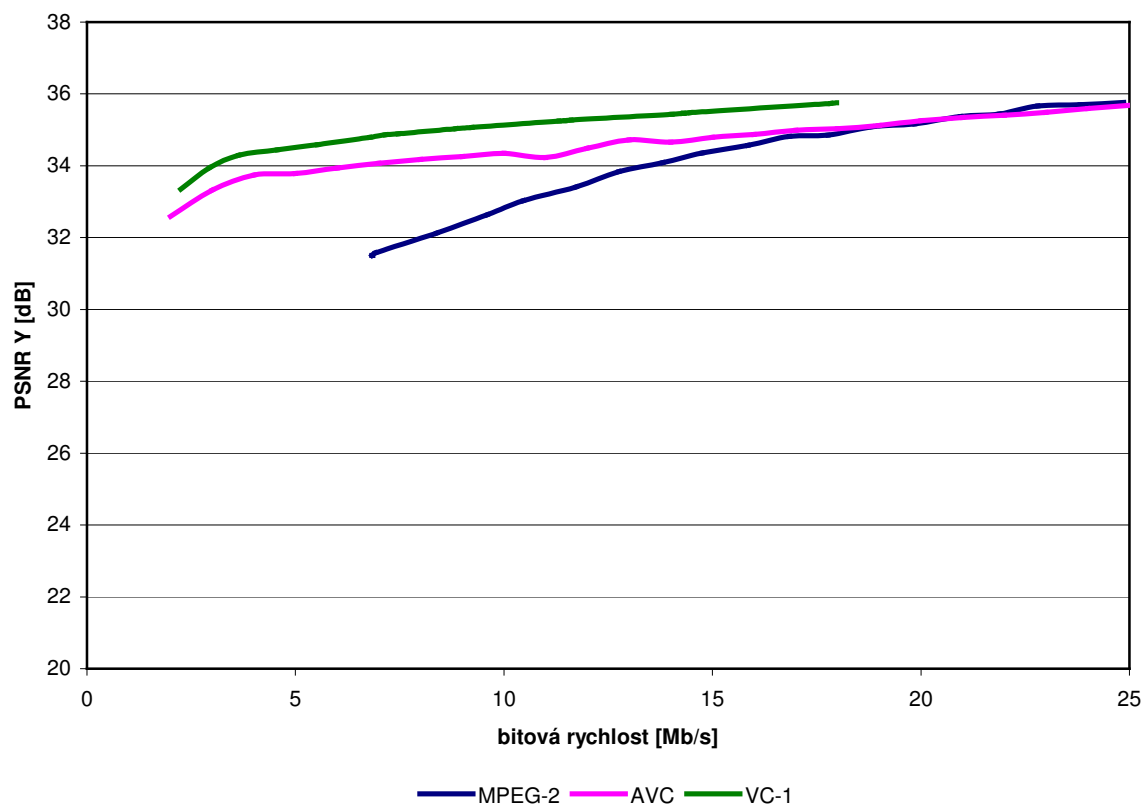




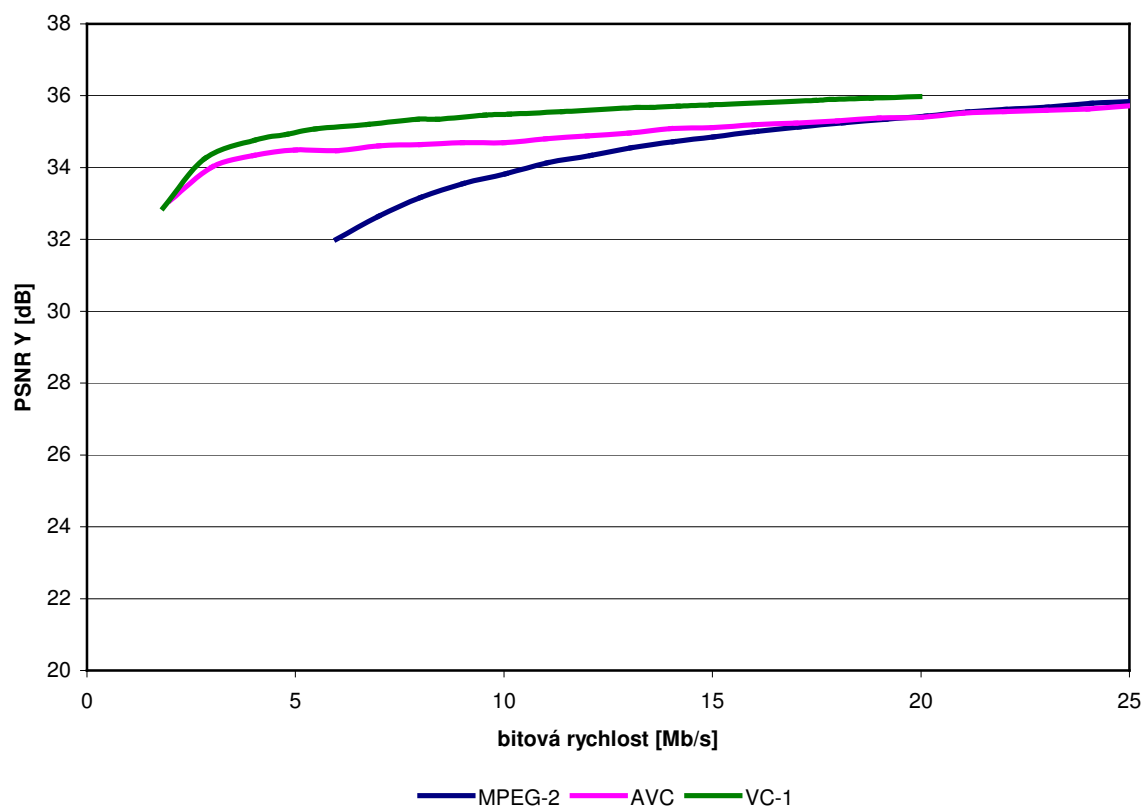
Obr. 45. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci ParkJoy.



Obr. 46. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci DucksTakeOff.



Obr. 47. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci InToTree.



Obr. 48. Srovnání závislostí PSNR jasové složky Y na bitové rychlosti pro sekvenci OldTownCross.

## 7.2.2 Závislosti indexu SSIM na bitové rychlosti

Následujících 5 grafů, viz obr. 49, 50, 51, 52, 53, zobrazuje srovnání algoritmů pro jednotlivé videosekvence. V tabulce 12 jsou pro příklad uvedeny hodnoty bitových rychlostí jednotlivých algoritmů pro komprimované videosekvence se stejnou hodnotou indexu SSIM.

U sekvence CrowdRun dosahoval nejlepších hodnot algoritmus AVC. Algoritmus VC-1 měl na celém společném rozsahu hodnoty indexu SSIM menší o cca 0,025 než AVC. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC a dosahoval menších hodnot než AVC. Rozdíl na nižších bitových rychlostech měl hodnotu až 0,1 a s rostoucí bitovou rychlostí se zmenšoval, až na cca 0,05 při nejvyšší hodnotě bitové rychlosti – 25Mb/s.

U sekvence ParkJoy byl průběh podobný jako u sekvence CrowdRun. Algoritmus VC-1 měl na celém společném rozsahu hodnoty indexu SSIM menší o cca 0,05 než AVC. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC a dosahoval menších hodnot než AVC. Rozdíl na nižších bitových rychlostech měl hodnotu 0,13 a s rostoucí bitovou rychlostí se zmenšoval, až na cca 0,07 při nejvyšší hodnotě bitové rychlosti – 25Mb/s.

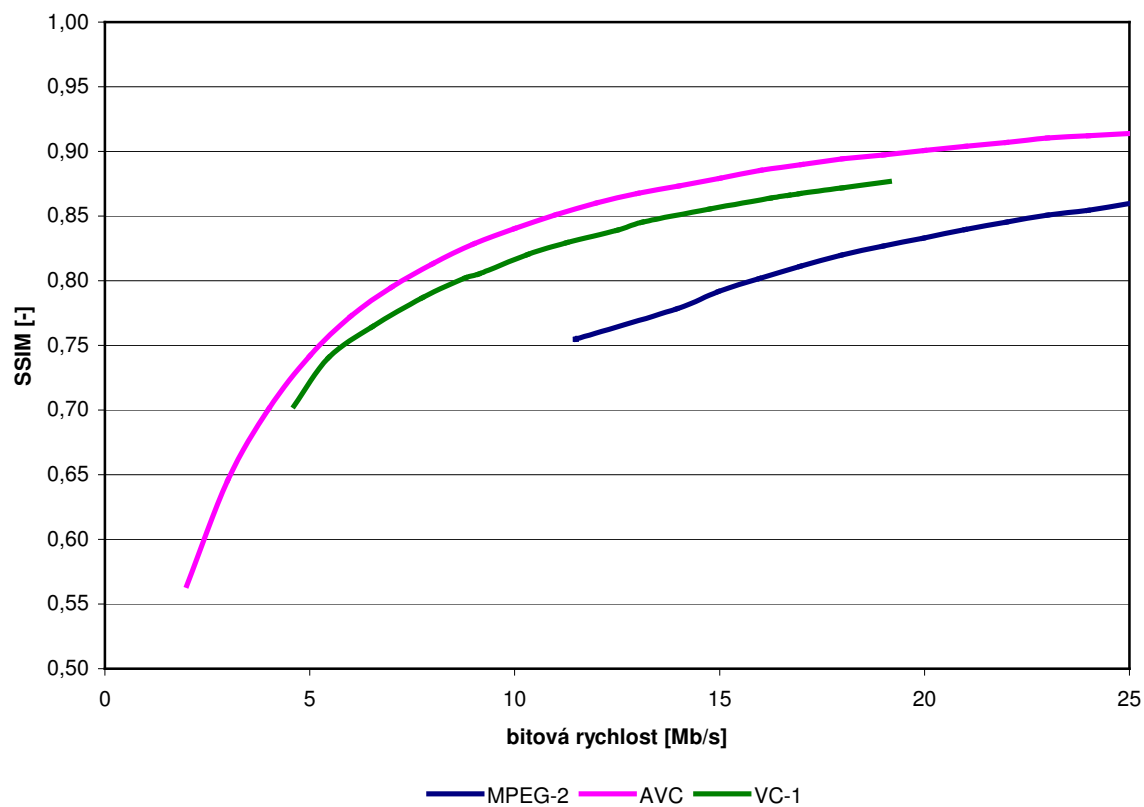
U sekvence DucksTakeOff byl průběh podobný jako u sekvence CrowdRun, ParkJoy. Algoritmus VC-1 měl na nižších hodnotách bitové rychlosti hodnoty indexu SSIM menší o cca 0,05 než AVC, na vyšších hodnotách bitové rychlosti pak rozdíl klesal až na cca 0,02. Algoritmus MPEG-2 měl podobný průběh jako algoritmus AVC a dosahoval menších hodnot než AVC. Rozdíl na nižších bitových rychlostech měl hodnotu 0,08 a s rostoucí bitovou rychlostí se zmenšoval, až na cca 0,05 při nejvyšší hodnotě bitové rychlosti – 25Mb/s.

U sekvence InToTree měl algoritmus VC-1 na celém společném rozsahu s AVC hodnoty větší o cca 0,03. Algoritmus MPEG-2 dosahoval v rozsahu hodnoty 6Mb/s až 14Mb/s menších než AVC, až o 0,05. S rostoucí bitovou rychlostí rozdíl klesal. Od 14Mb/s do 25Mb/s dosahoval algoritmus MPEG-2 vyšších hodnot než AVC.

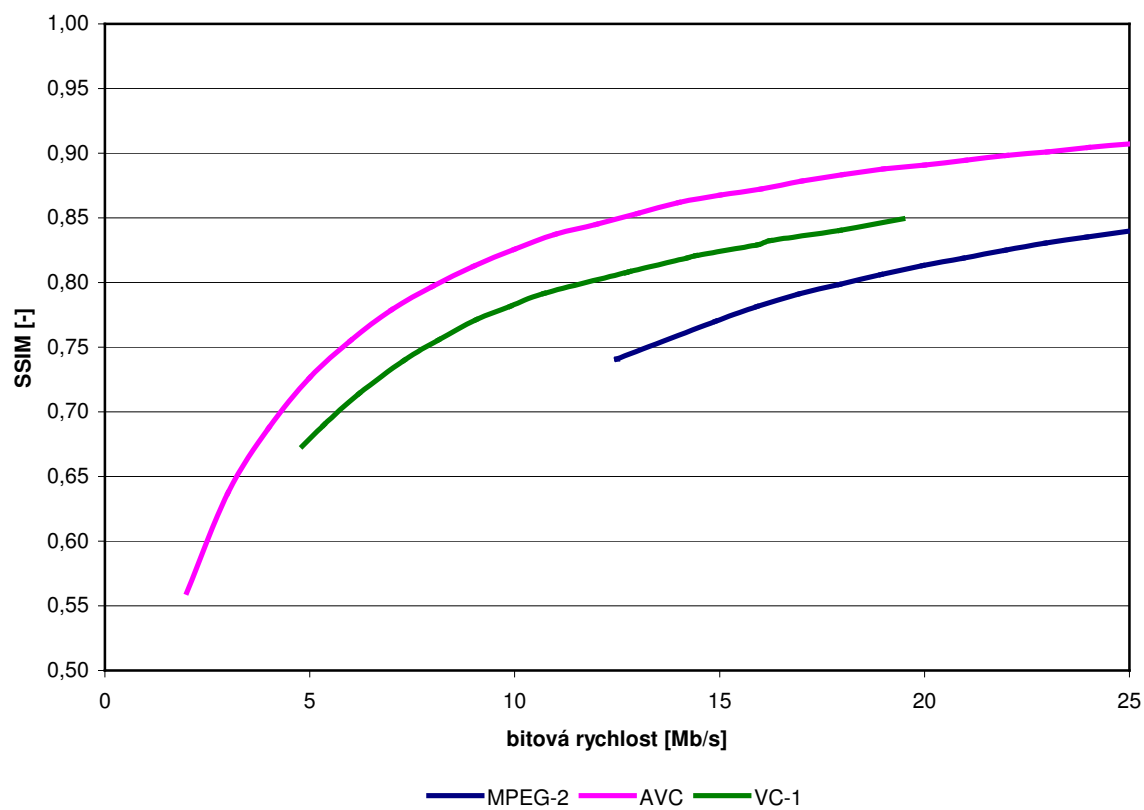
U sekvence OldTownCross dosahoval algoritmus VC-1 na celém společném rozsahu s AVC hodnot vyšších. Algoritmus MPEG-2 dosahoval v rozsahu hodnoty 6Mb/s až 14Mb/s menších než AVC, až o 0,03. S rostoucí bitovou rychlostí rozdíl klesal. Od 14Mb/s do 25Mb/s byly průběhy algoritmů MPEG-2 a AVC totožné.

| Sekvence     | Obtížnost kódování | SSIM [-] | Bitová rychlost [Mb/s] |      |        |
|--------------|--------------------|----------|------------------------|------|--------|
|              |                    |          | AVC                    | VC-1 | MPEG-2 |
| CrowdRun     | obtížná            | 0,8      | 7,2                    | 8,7  | 15,9   |
| DucksTakeOff | obtížná            | 0,8      | 9,5                    | 11,4 | 19     |
| ParkJoy      | obtížná            | 0,8      | 8,2                    | 11,9 | 18     |
| InToTree     | snadná             | 0,85     | 18                     | 8    | 16     |
| OldTownCross | snadná             | 0,85     | 2,2                    | 1,8  | 8      |

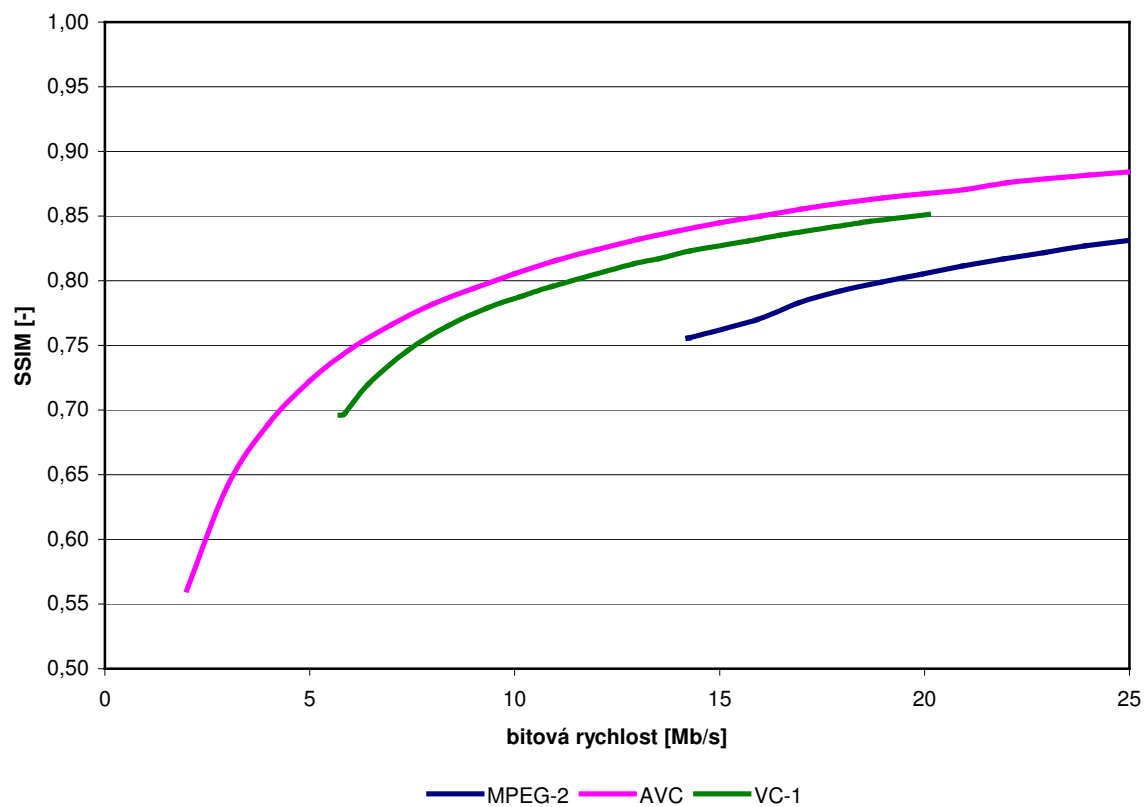
Tab. 12. Srovnání bitové rychlosti algoritmů při shodné kvalitě (SSIM/1080p).



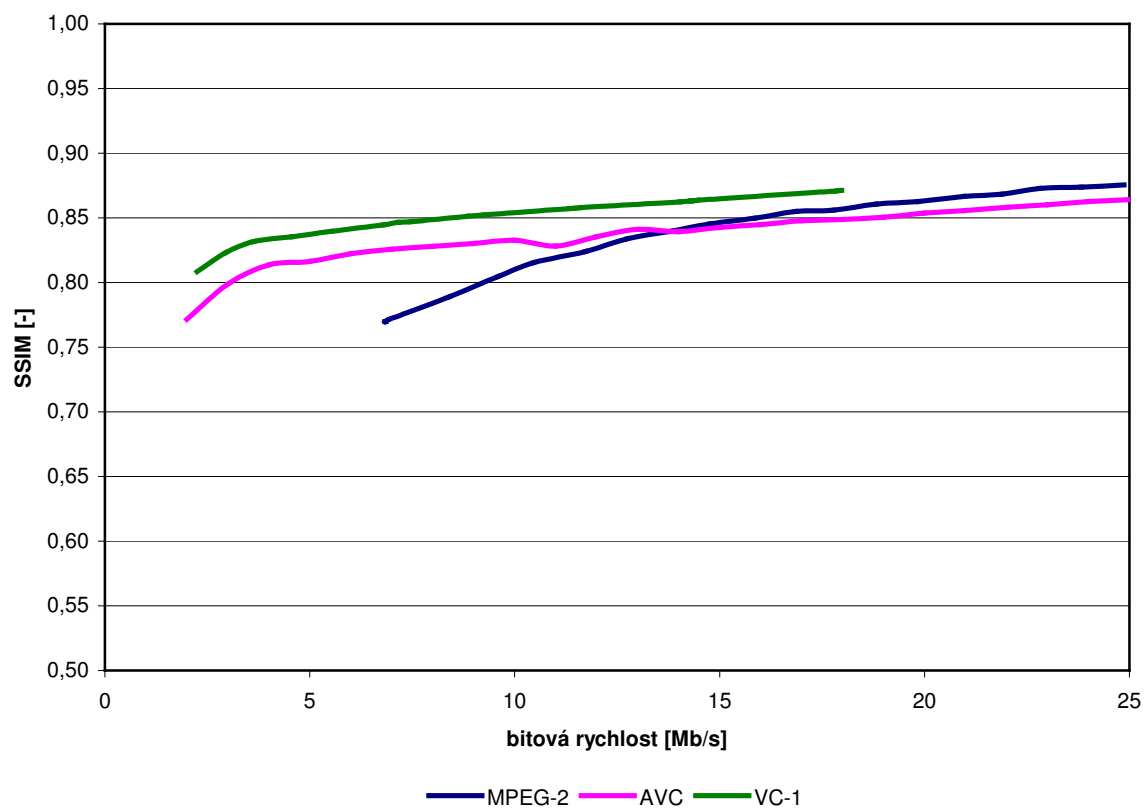
Obr. 49. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci CrowdRun.



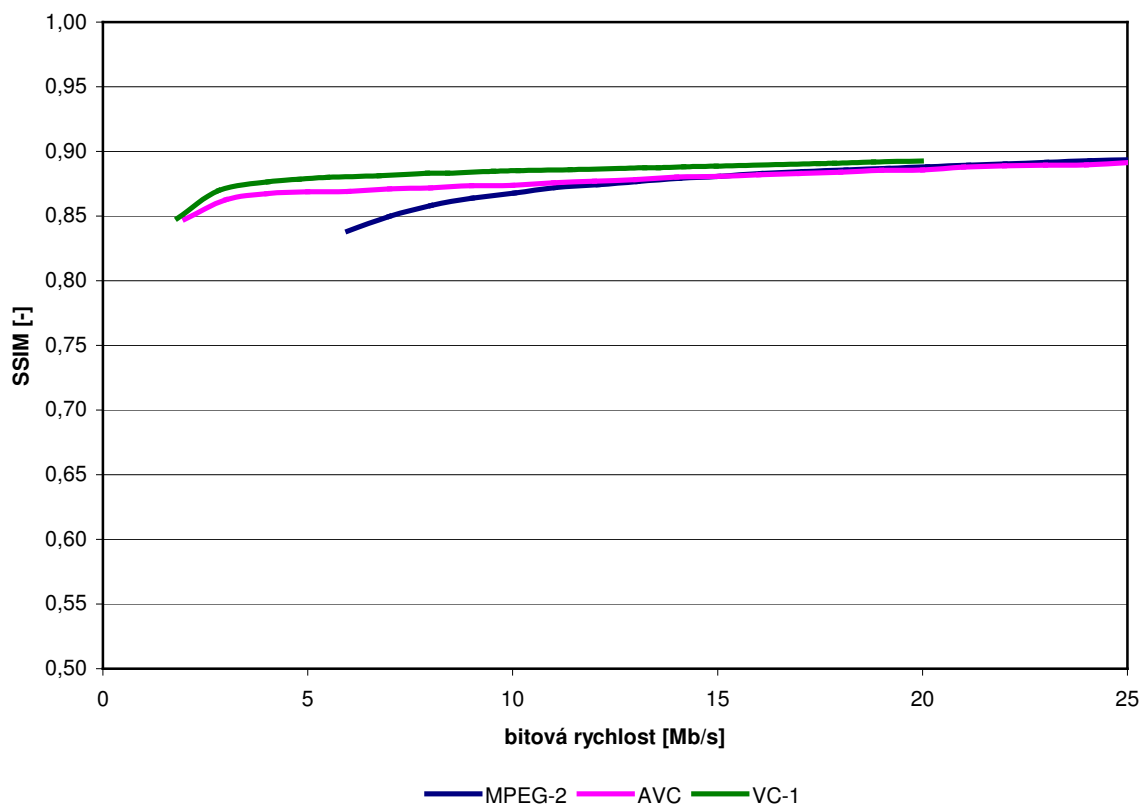
Obr. 50. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci ParkJoy.



Obr. 51. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci DucksTakeOff.



Obr. 52. Srovnání závislostí indexu SSIM na bitové rychlosti pro sekvenci InToTree.



Obr. 53. Srovnání závislosti indexu SSIM na bitové rychlosti pro sekvenci OldTownCross.

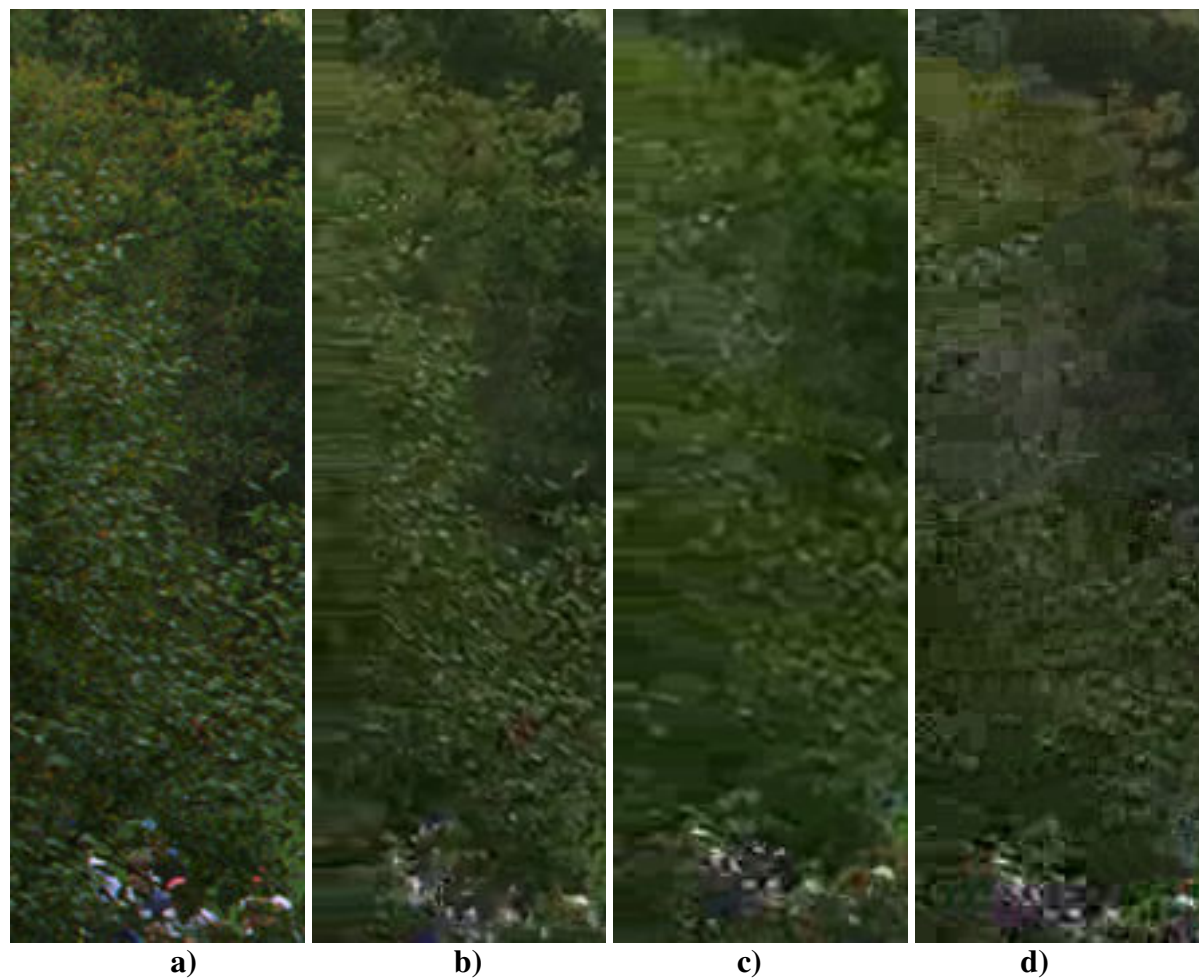
### 7.1.3 Porovnání subjektivní kvality

Jedná se o srovnání originálního, nekomprimovaného snímku se snímky komprimovanými jednotlivými algoritmy, viz obr. 54. Pro porovnání subjektivní kvality byl vybrán, stejně jako u formátu 720p, snímek 75 ze sekvence ParkJoy, komprimované sekvence dosahovaly přibližně stejné hodnoty PSNR = 28 dB.



Obr. 54. Originální snímek s vybranými oblastmi pro porovnání – 1080p.

Oblast 1, viz obr. 55 – Algoritmus AVC vykazuje částečné rozmazání obrazu na levém okraji oblasti. U algoritmu VC-1 je rozmazání obrazu o něco větší než u AVC, vytváří se blokové artefakty. U algoritmu MPEG-2 se vytváří blokové artefakty a mírně rozmazán.



Obr. 55. Porovnání kvality snímků a) originální b) AVC c) VC-1 d) MPEG-2.

## 8 Závěr

V práci byly popsány nástroje a algoritmy tří, v současné době nejpoužívanějších, komprimačních standardů pro videozáznamy a to MPEG-2 Video, MPEG-4 AVC a VC-1. Využití těchto standardů je od televizního vysílání, kde dominuje MPEG-2 Video a rozšiřující se MPEG-4 AVC po standardy záznamů u přenosových médií – MPEG-2 Video pro DVD media, MPEG-4 AVC a VC-1 jako standardy pro HD DVD a Blu-Ray discích. V práci také byly porovnány jednotlivé algoritmy, co do jejich výkonnosti a subjektivní kvality komprimovaných videosekvencí ve vysokém rozlišení. Pro měření kvality byla místo standardně využívané metody špičkového poměru signálu k šumu PSNR použita i metoda strukturální podobnosti SSIM, které je ovšem bližší lidskému subjektivnímu dojmu než PSNR.

Z porovnání výkonnosti vyplývá, že ze tří algoritmů se chová nejhůře MPEG-2 Video, a to u všech testovaných videosekvencí a rozlišení. Nejlépe se naopak choval MPEG-4 AVC, který pro obě testovaná rozlišení a videosekvence se složitým pohybem ve scéně s velkým množstvím detailů dosahoval nejlepších hodnot. U videosekvencí s minimálním pohybem ve scéně se pro rozlišení 1280x720 choval lépe MPEG-4 AVC a naopak u rozlišení 1920x1080 zase VC-1.

I přes poměrně velký rozsah obsahuje práce jen základní analýzu nástrojů používaných pro zpracování obrazu. Jednotlivé algoritmy by bylo možné doplnit o další speciální nástroje, týkající se zpracování obrazu, aby bylo jejich porovnání komplexní. Práci by bylo možné rozšířit práci o další komprimační algoritmy, jelikož současná práce popisuje jen zlomek z dostupných komprimačních algoritmů.



## Seznam literatury

- [1] VÍT, V. *Televizní technika, přenosové barevné soustavy*. Praha: BEN technická literatura, 1997. 719 stran. ISBN 80-86056-04-X.
- [2] WIKIPEDIE. *MPEG-2* [online]. [cit. 19. 11. 2009]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/MPEG-2>
- [3] PLANINŠIC, P., MOHORKO, J., CUCEJ, Ž. *Komprimiranje slik* [online]. [cit. 19. 11. 2009]. Dostupný z WWW: <http://www.sparc.uni-mb.si/diplome/s6i.pdf>
- [4] TOMAN J., PROCHÁZKA I. *Kompresní technologie DVB* [online]. [cit. 19. 11. 2009]. Dostupný z WWW: <http://www.ceskatelevize.cz/specialy/digict/kapitola4.htm>
- [5] HAGLUND, L. *The SVT High Definition Multi Format Test Set*. [online]. [cit. 18. 04. 2010]. Dostupný z WWW: <ftp://vqeg.its.bldrdoc.gov>
- [6] HARTE, L. *MPEG Profiles* [online]. [cit. 21. 11. 2009]. Dostupný z WWW: [http://www.althos.com/IPTVArticles/iptvmagazine\\_2006\\_11\\_MPEG.htm](http://www.althos.com/IPTVArticles/iptvmagazine_2006_11_MPEG.htm)
- [7] *MPEG compression technique* [online]. [cit. 21. 11. 2009]. Dostupný z WWW: [http://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg\\_tech.html](http://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg_tech.html)
- [8] JAHODA, R. *Kodeky tajemství zbavené* [online]. [cit. 21. 11. 2009]. Dostupný z WWW: [http://www.tvfreak.cz/art\\_doc-373A9DA2913B7BD3C125727C00592A37.html](http://www.tvfreak.cz/art_doc-373A9DA2913B7BD3C125727C00592A37.html)
- [9] RICHARDSON, I. E. G. *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. Chichester: Wiley, 2003.
- [10] SLANINA, M. *Methods and Tools for Image and Video Quality Assessment*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 80 s.
- [11] HAGLUND, L. *The SVT High Definition Multi Format Test Set* [online]. [cit. 29. 03. 2010]. Dostupný z WWW: [tech.ebu.ch/docs/hdvtv/svt-multiformat-conditions-v10.pdf](http://tech.ebu.ch/docs/hdvtv/svt-multiformat-conditions-v10.pdf)
- [12] WIKIPEDIE. *PSNR* [online]. [cit. 29. 03. 2010]. Dostupný z WWW: <http://en.wikipedia.org/wiki/PSNR>
- [13] WIKIPEDIE. *SSIM* [online]. [cit. 29. 03. 2010]. Dostupný z WWW: <http://en.wikipedia.org/wiki/SSIM>
- [14] BROOKS, A., PAPPAS N. *PappasStructural similarity quality metrics in a coding context: exploring the space of realistic distortions* [online]. [cit. 01. 04. 2010]. Dostupný z WWW: [www.eecs.northwestern.edu/~pappas/papers/brooks\\_hvei06.pdf](http://www.eecs.northwestern.edu/~pappas/papers/brooks_hvei06.pdf)

- [15] SHETH, B. *Structural Similarity Index* [online]. [cit. 01. 04. 2010]. Dostupný z WWW: [www-ee.uta.edu/Dip/Courses/EE5359/Project\\_proposals2008/Bishwa/final%20presentation\\_mm.ppt](http://www-ee.uta.edu/Dip/Courses/EE5359/Project_proposals2008/Bishwa/final%20presentation_mm.ppt)
- [16] PROJROCK, D. *Function to compute the PSNR between two YUV videos* [online]. [cit. 01. 04. 2010]. Dostupný z WWW: <http://www.mathworks.co.uk/matlabcentral/fileexchange/12455-psnr-of-yuv-videos>
- [17] LEFUNGUSSSIM 0.24 Dll [online]. [cit. 01. 04. 2010]. Dostupný z WWW: <http://perso.wanadoo.fr/reservoir/dl/SSIM-0.24.rar>
- [18] WANG, Z. *The SSIM Index for Image Quality Assessment* [online]. [cit. 01. 04. 2010]. Dostupný z WWW: <http://www.cns.nyu.edu/~zwang/>
- [19] KALVA, H., LEE, J. *The VC-1 and H.264 Video Compression Standards for Broadband Video Services*. Springer, 2008
- [20] GOLDMAN, M. *A Comparision of MPEG-2 Video, MPEG-4 AVC, and SMPTE VC-1* [online]. [cit. 17. 04. 2010]. Dostupný z WWW: [http://video.ldc.lu.se/pict/WM9V-MP4AVC-MP2V\\_comparison-Goldman.pdf](http://video.ldc.lu.se/pict/WM9V-MP4AVC-MP2V_comparison-Goldman.pdf)
- [21] HANDBRAKE. *x264 Options in HandBrake* [online]. [cit. 18. 04. 2010]. Dostupný z WWW: <http://trac.handbrake.fr/wiki/x264Options>

# Seznam příloh

## A DVD médium

- A.1 Databáze komprimovaných vidoesekvencí MPEG-2 (database\ MPEG2\...)
- A.2 Databáze komprimovaných vidoesekvencí AVC (database\AVC\...)
- A.3 Databáze komprimovaných vidoesekvencí VC-1 (database\ VC1\...)